

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

До захисту допущено:

Завідувач кафедри

Стіренко С.Г. _____

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерні системи та мережі»
спеціальності 123 «Комп'ютерна інженерія»
на тему: «Додаток для обробки інтернет замовлень»**

Виконав:

студент IV курсу, групи ІО-61
Кузьменко Богдан Сергійович _____

Керівник:

Асистент

Регіда Павло Геннадійович _____

Консультант з нормоконтролю:

Професор д.т.н.

Сімоненко Валерій Павлович _____

Рецензент:

Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467800.003 ПЗ	Пояснювальна записка	48	
3	A4	ІАЛЦ.467800.003 ОА	Опис альбому	1	
4	A4	ІАЛЦ.467800.003 ТЗ	Технічне завдання	1	
5	A4	ІАЛЦ.467800.004 Д1	Додаток 1	1	
6	A4	ІАЛЦ.467800.005 Д2	Додаток 2	1	
7	A4	ІАЛЦ.467800.006 Д3	Додаток 3	1	
8	A4	ІАЛЦ.467800.007 Д4	Додаток 4	14	

				ІАЛЦ.467800.003		
	ПБ	Підп.	Дата	Відомість дипломного проєкту	Лист	Листів
Розробн.	Кузьменко Б.С.				1	1
Керівн.	Регіда П.Г.				КПІ ім. Ігоря Сікорського Каф. ОТ Гр. ІО-61	
Консульт.						
Н/контр.	Сімоненко В.П.					
Зав.каф.	Стіренко С.Г.					

**Пояснювальна записка
до дипломного проєкту
на тему: «Додаток для обробки інтернет замовлень»**

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп’ютерна інженерія»

Освітньо-професійна програма «Комп’ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Стіренко С.Г. _____

«___» _____ 2020 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Кузьменко Богдану Сергійовичу

1. Тема проєкту «Додаток для обробки інтернет замовлень», керівник проєкту асистент Регіда Павло Геннадійович, затверджені наказом по університету від «07» травня 2020 р. № 1081-С
2. Термін подання студентом проєкту: 15 червня 2020 р.
3. Вихідні дані до проєкту: данні у формі таблиці, інформація, що надходить з Телеграму.
4. Зміст пояснювальної записки: огляд існуючих рішень, середовищ розробки, варіанти взаємодії.
5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): 28 рисунки, 3 блок-схеми.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко Валерій Павлович Професор д.т.н.		

7. Дата видачі завдання: 09.11.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Затвердження теми роботи</i>	<i>01.09.2019</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>04.10.2019</i>	
3.	<i>Розробка архітектури та загальної структури систем</i>	<i>03.11.2019</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>05.12.2019</i>	
5.	<i>Програмна реалізація системи</i>	<i>04.02.2020</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>07.03.2020</i>	
7.	<i>Захист програмного продукту</i>	<i>03.04.2020</i>	
8.	<i>Передзахист</i>	<i>26.05.2020</i>	
9.	<i>Захист</i>	<i>15.06.2020- 20.06.2020</i>	

Студент

Богдан КУЗЬМЕНКО

Керівник

Павло РЕГІДА

АНОТАЦІЯ

В бакалаврській дипломній роботі реалізовано додаток для обробки інтернет-замовлень, призначений для використання невеликими магазинами як аналог сайту або інших платформ. За основу був обраний відомий та розповсюджений додаток для обміну повідомленнями – Telegram, мова програмування – Python 3.8+, база даних – Google Sheets. Для клієнта програма дозволяє обрати та замовити товар, для власника – зручніше вести облік, редагувати наявний товар та одразу мати зв'язок із клієнтом.

АННОТАЦИЯ

В бакалаврской дипломной работе реализовано приложение для обработки интернет-заказов, предназначенное для использования небольшими магазинами как аналог сайта или других платформ. В качестве основания был взят известный и популярный мессенджер – Telegram, язык программирования – Python 3.8+, база данных – Google Sheets. Для клиента программа позволяет выбрать и заказать товар, для собственника – удобнее вести учёт, редактировать товар в наличии и сразу иметь связь с клиентом.

ANNOTATION

In the bachelor's thesis, an application for processing online orders is implemented, intended for use by small shops as an analogue of a site or other platforms. The well-known and popular messenger - Telegram, the programming language - Python 3.8+, the database - Google Sheets was taken as the basis. For the client, the program allows to select and order goods, for the owner it is more convenient to keep records, edit the goods in stock and immediately have a connection with the client.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	2
ВСТУП	3
1 КОНЦЕПЦІЯ БОТУ ТА ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ КУПІВЛІ ЧЕРЕЗ МЕРЕЖУ ІНТЕРНЕТ	5
1.1 Вступ	5
1.2 Порівняння та огляд обраних платформ для здійснення інтернет-покупок	6
1.3 Огляд ботів в цілому та можливостей до їх адаптації під різноманітні потреби	14
1.4 Огляд та аналіз основ і допоміжних засобів для створення боту	19
ВИСНОВКИ ДО РОЗДІЛУ 1	28
2 ОГЛЯД ПРИЧИН, СЕРЕДОВИЩА, МОВ ПРОГРАМУВАННЯ, ДОПОМІЖНИХ БІБЛІОТЕК ТА ЗАСОБІВ ДЛЯ СТВОРЕННЯ І ПІДТРИМКИ РОБОТИ ЗАСТОСУНКУ	29
2.1 Аналіз необхідності створення власного боту	29
2.2 Вибір мови програмування	30
2.3 Аналіз допоміжних засобів для створення	32
2.4 Огляд середовища розробки	34
ВИСНОВКИ ДО РОЗДІЛУ 2	36
3 НАПИСАННЯ ВЛАСНОГО БОТУ	37
3.1 Загальний огляд проекту	37
ВИСНОВКИ ДО РОЗДІЛУ 3	48
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	50

					ІАЛЦ.467800.003 ПЗ										
					Додаток для обробки інтернет замовлень										
Зм.	Аркуш	№ Докум.	Підпис	Дата											
	Розробив	Кузьменко Б. С.													
	Перевірив	Регіда П. Г.													
	Т. Контр.									Аркуш 1		Аркушів 48			
										КПІ ім. Сікорського ФІОТ Група ІО-61					
	Н. контр.	Сімоненко В. П.													
	Затвердив	Регіда П. Г.													

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ОС	–	Операційна система
ПК	–	Персональний комп'ютер
ПО	–	Програмне забезпечення
Соц. мережа	–	Соціальна мережа
Бот	–	Пограмне забезпечення, що імітує поведінку живої людини, програмується для виконання різноманітних завдань
IDLE	–	(Integrated Development and Learning Environment) Середовище розробки
Скрипт (Сценарій)	–	Програма, яка є складовою проекту.
API	–	(Application programming interface) Інтерфейс для взаємодії

					<i>ІАЛЦ. 467800.003 ПЗ</i>	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

Вступ

Торгівля як процес – починає свою історію ще з кам'яного віку. Виникла вона для вирішення проблеми перевиробництва товарів шляхом їх обміну на щось інше. Опустивши багато деталей з історії розвитку цього явища - перейдемо одразу до моменту, коли один товар почали обмінювати не на інший товар, а на засіб підтвердження свого статку – чеки. Чеками називали певні квитанції, за які у праобразу банку можна було отримати відповідну кількість золота або іншого дорогоцінного металу.

Зростання популяції планети Земля було та залишається неспівставним з ростом об'ємів видобутку дорогоцінних металів таких як золото, срібло і т.д.

Для забезпечення більшої кількості людей можливістю вести торгівлю – необхідна більша кількість чеків. З причин описаної вище ситуації з видобутком дорогоцінних металів, довелося знижувати купівельну спроможність одного такого чеку. З плином часу було прийняте рішення про відв'язку чеків від дорогоцінного металу – так з'явилися перші гроші.

Не менш важливим для нас моментом в історії людства можна вважати появу об'єднань людей за навичками та інтересами для ведення спільної торгівлі – це стало праобразом сучасних торгівельних мереж та магазинів.

Повертаючись до наших днів, важко уявити існування будь-якого магазину без можливості зробити покупку через мережу інтернет. У тому чи іншому вигляді, онлайн-платформа для купівлі товару – значна перевага у порівнянні з її відсутністю. Враховуючи різноманітні служби доставки, можливість дистанційного вибору та оформлення замовлення – дозволяють повністю виключити необхідність живої присутності людини у цьому процесі.

					ІАЛЦ. 467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

Таким чином буденні справи займають менше часу та стають набагато простішими. Окрім спрощення певних процесів, онлайн засоби для купівлі товарів надають змогу безпечно існувати у періоди пандемій та спалахів різноманітних захворювань, коли краще зменшити кількість контактів з іншими людьми. Також це значно допомагає літнім людям та людям із обмеженими можливостями.

Об'єктом дипломної роботи є додаток для обробки інтернет замовлень – бот. Вибір зумовлений власним бажанням та зацікавленістю, окрім цього, боти як явище стають все більш актуальними та розповсюдженими.

Предметом дослідження є написання процедурного бота для обробки інтернет замовлень, на базі додатку для обміну повідомленнями `Telegram`, з використанням допоміжних сервісів Google Sheets та Google Photos.

Метою роботи є:

1. Написання власного боту для обробки інтернет-замовлень та здійснення покупок онлайн.

Завданнями роботи є:

1. Огляд існуючих засобів для здійснення інтернет-покупок, їх переваги та недоліки для магазинів різного масштабу. Огляд ботів в цілому та їх адаптацію під дану предметну галузь
2. Аналіз причин, середовища, мови програмування, обраних бібліотек та допоміжних засобів для створення і підтримки роботи застосунку.
3. Написання власного боту.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1.

КОНЦЕПЦІЯ БОТУ ТА ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ КУПІВЛІ ЧЕРЕЗ МЕРЕЖУ ІНТЕРНЕТ

1.1 Вступ

Історія ідеї створення всесвітньої мережі інтернет бере свій початок з 1962-го року. Створення та запуск цієї мережі поділив історію людства на «до» і «після», створивши величезну кількість нових можливостей, реалізувавши безліч, неможливих на той час, концепцій та зробивши життя людей набагато швидшим, а отже і простішим.

Одною із таких концепцій та можливостей є простіша дистанційна торгівля, з можливістю перегляду зображень товару, його опису та власне замовлення. Із розвитком комп'ютерних та мобільних технологій, такий спосіб вести торгівлю, нарешті, став можливим – це значно полегшило та прискорило товарообіг у світі.

З плином часу, платформ для здійснення такого роду покупок стало чимало. Обрати найбільш придатний для себе формат тепер є можливим з досить великої кількості запропонованих.

Серед актуальних та найбільш розповсюджених варіантів інтернет-магазинів, для порівняння, були обрані наступні:

- Інтернет магазин на основі веб-сайту.
- Інтернет магазин на основі сторінки у соц. мережі.
- Інтернет магазин на основі додатку для ПК або мобільного пристрою.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Порівняння та огляд обраних платформ для здійснення інтернет-покупок

1.2.1 Інтернет магазин на основі веб-сайту

Є найбільш розповсюдженим і чудово підійде для великих та середніх магазинів, коли мова йде про велику кількість товарів та клієнтів.

Зазвичай, використовується стаціонарно, тоб то користувач перебуває за великим монітором персонального комп'ютера або ноутбуку. Використання з мобільних пристроїв з меншими дисплеями також можливе, та є менш зручним в силу не завжди якісної адаптації сайту зі сторони розробників.

Переваги:

- Можливість алаштування розробниками дизайну під різні пристрої та під будь-який смак.
- Широкий вибір доступного функціоналу (Особистий кабінет, новини, реклама і т.д.)
- Зазвичай, більше можливостей у порівнянні з іншими реалізаціями.

Недоліки:

- Відносно складний і дорогий процес створення, налаштування, адаптації та підтримки.
- Якщо мова йде про повноцінний сайт – необхідність купівлі домену та оренди серверу.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

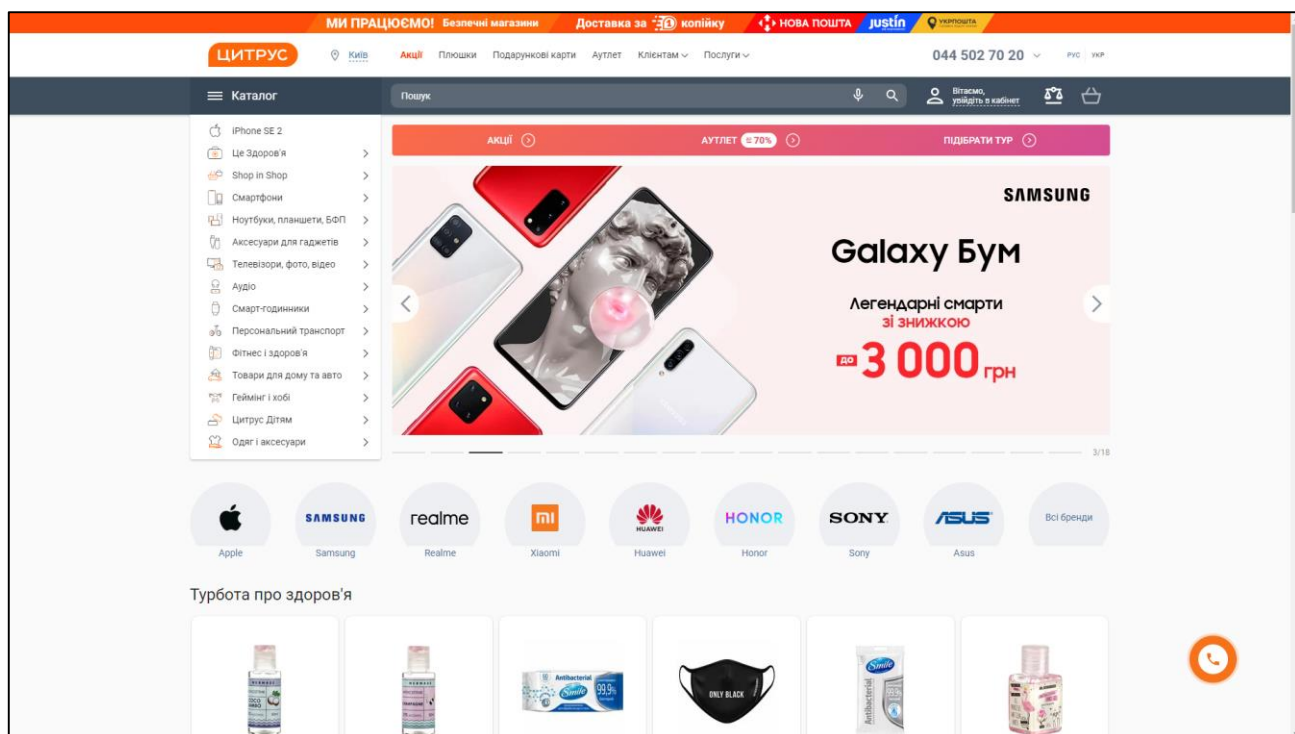


Рис. 1.1. Інтернет магазин на основі веб-сайту, адаптований під великий монітор ПК або ноутбуку.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

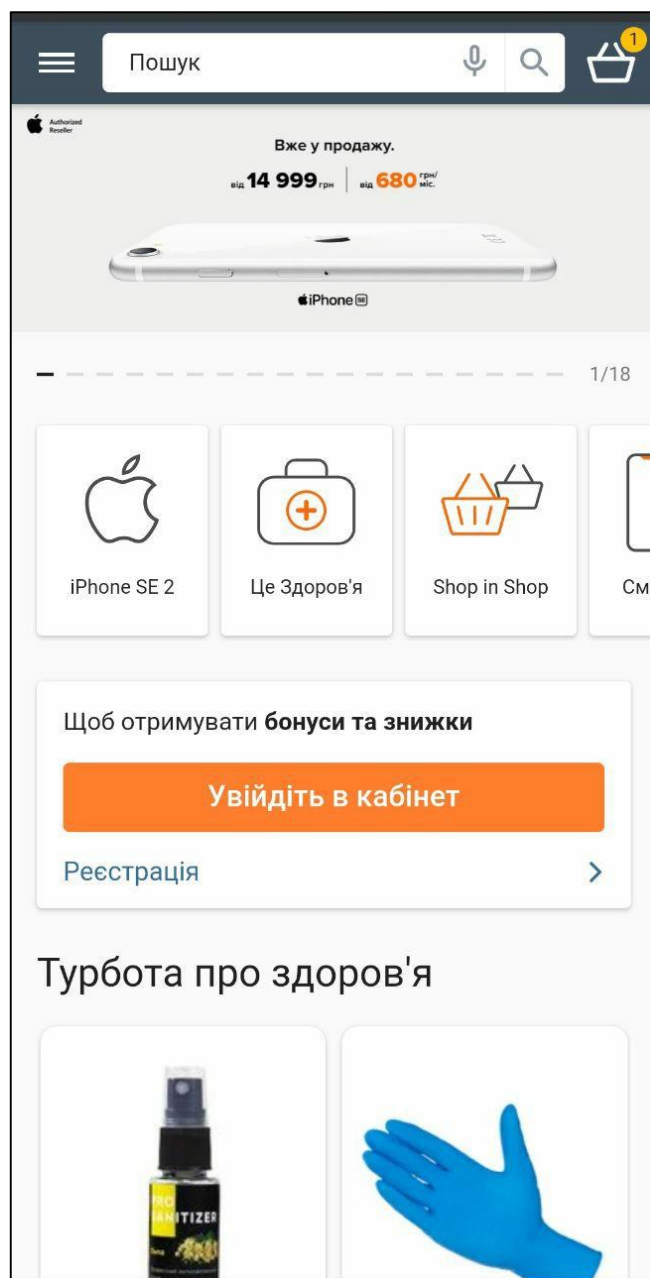


Рис. 1.2. Інтернет магазин на основі веб-сайту, адаптований під екран мобільного пристрою.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.2 Інтернет магазин на основі сторінки у соціальній мережі

Сторінки у таких соціальних мережах як Facebook або Instagram – досить поширені серед малих та середніх магазинів.

Такий формат дозволяє власникам безкоштовно «розмістити» свій магазин у мережі та здійснювати торгівлю.

Переваги:

- Розповсюдженість даних мереж сприяє легкому доступу до подібних інтернет магазинів та простій комунікації із замовником.
- Безкоштовне розміщення та використання, просте створення.
- Відсутність необхідності замовляти розробку, підтримку і т.д.
- Може виступати у якості реклами.

Недоліки:

- Непристосованість платформ до такого роду діяльності обмежує функціонал.
- Певні юридичні обмеження.
- Недоступність для тих, хто не є користувачами обраної соц. мережі.
- Можливе несерйозне ставлення до магазину або закладу.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

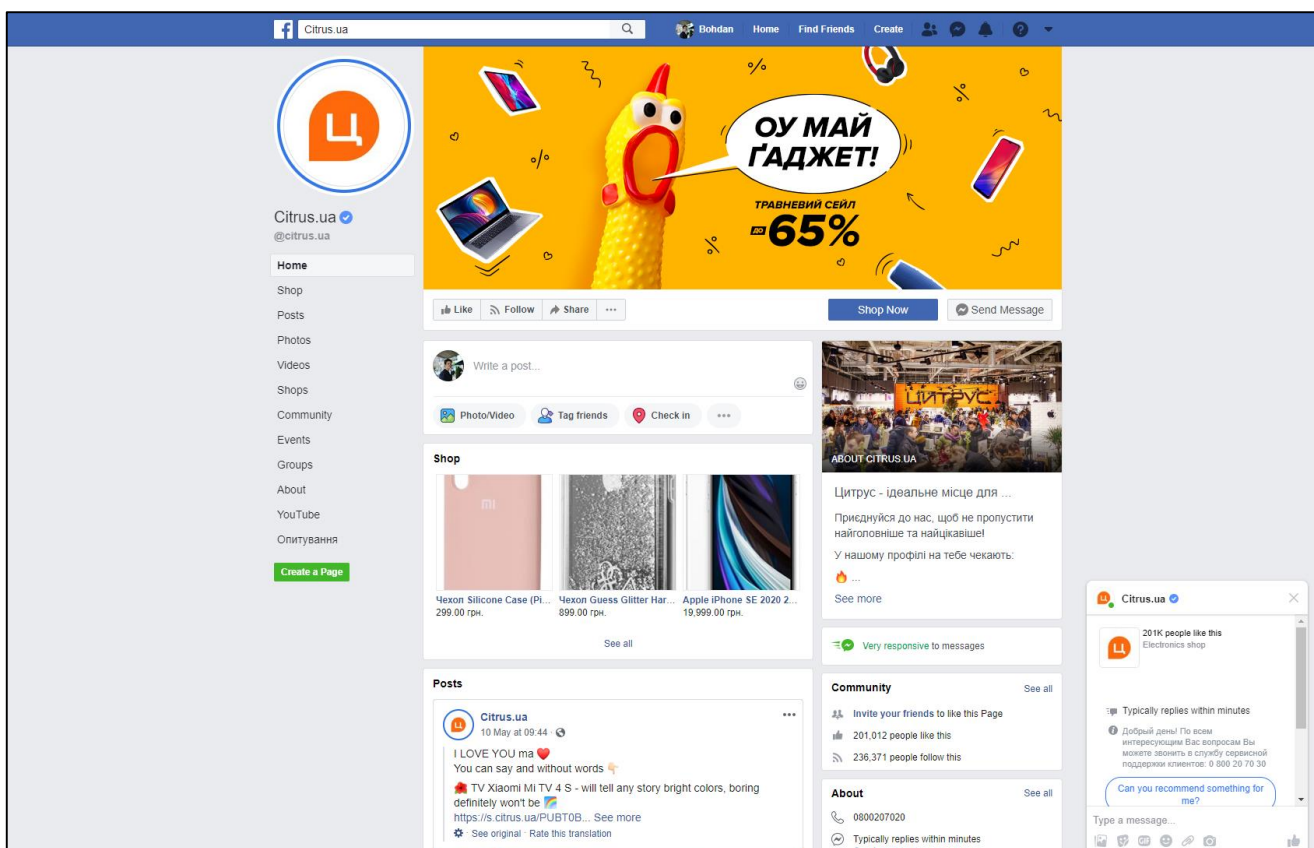


Рис. 1.3. Інтернет магазин на основі сторінки у соціальній мережі, адаптований під великий монітор ПК або ноутбуку.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

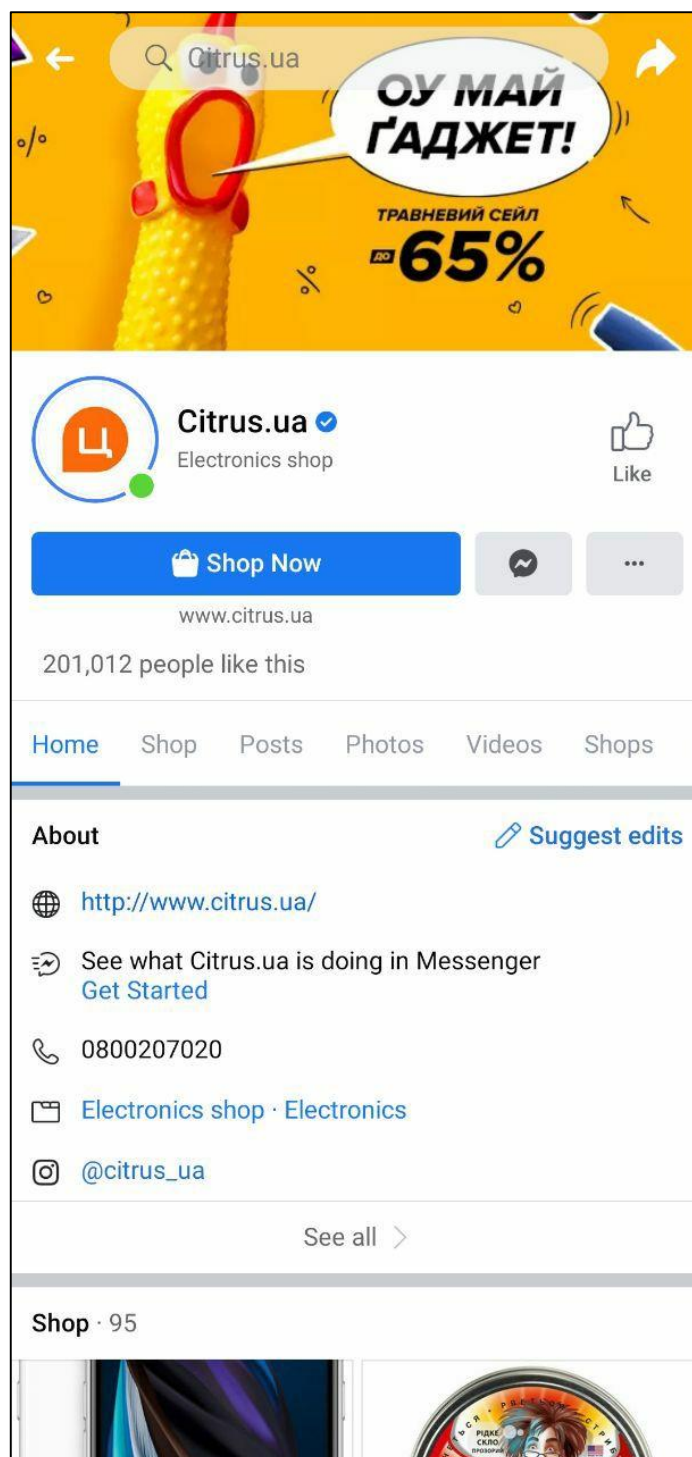


Рис. 1.4. Інтернет магазин на основі сторінки у соціальній мережі, адаптований під екран мобільного пристрою.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.3 Інтернет магазин - додаток для смартфона або комп'ютера

Не менш поширений варіант ніж попередні, та через відносно складне створення та підтримку – ним, зазвичай, не користуються малі магазини.

Переваги:

- Максимальна адаптація під конкретний тип пристроїв.

Недоліки:

- Необхідність окремої розробки та підтримки різних версій додатку для різних ОС.
- Користувача може відлякнути необхідність встановлення окремого додатку.
- Потенційна захащеність списку програм на пристрої.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

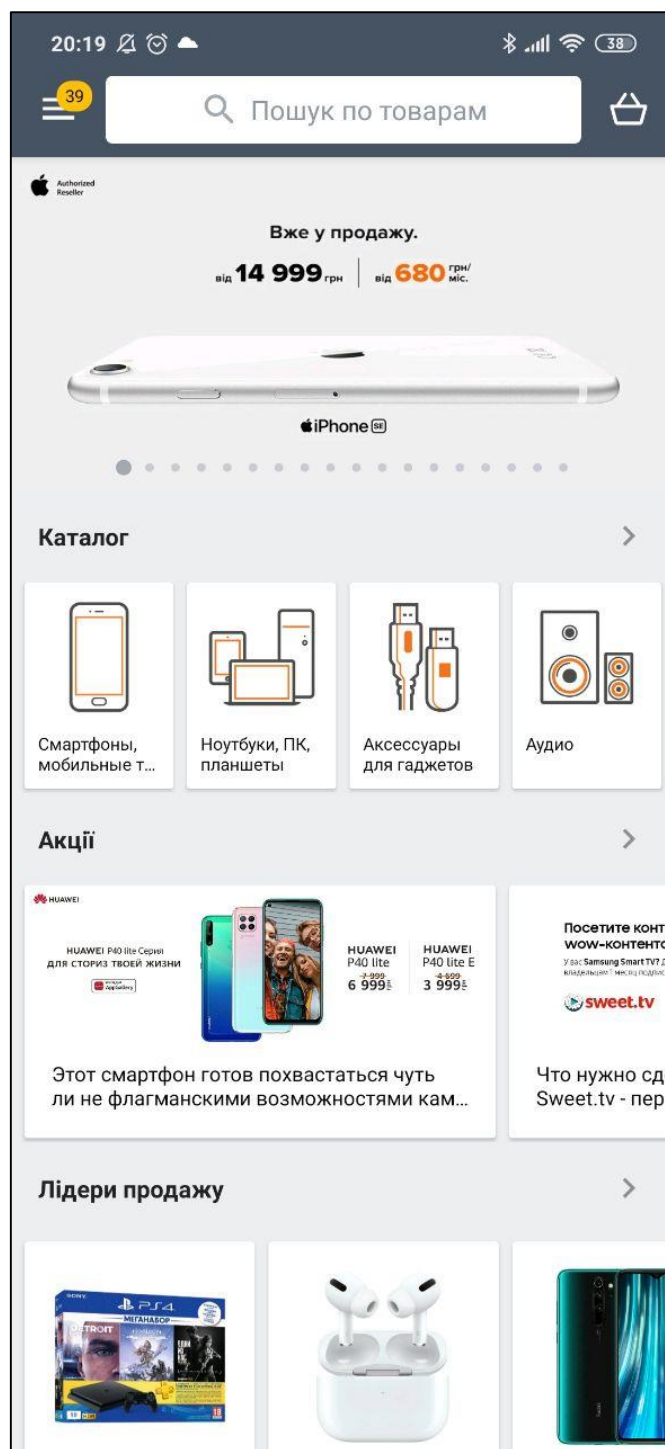


Рис. 1.5. Інтернет магазин – додаток для смартфона.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Огляд ботів в цілому та можливостей до їх адаптації під різноманітні потреби

Людство здавна мріяло про штучних «розумних» помічників які полегшували б життя виконуючи замість людей їх рутинні справи. В певному сенсі, реалізація саме такого функціоналу і закладена у поняття «Бот».

Звісно, мова не йде про щось справді розумне, адже навіть під поняттям «штучний інтелект» ховаються певні причинно-наслідкові алгоритми.

Фактично, бот – узагальнююча назва для певного набору інструкцій різної складності та реалізації, що виконують різні спеціальні дії, зазвичай не пов’язані з фізичною працею.

Відгалуженням від основного поняття стали чат-боти – програмно-надбудови в додатках для листування, для віддаленого виконання певних дій, наприклад купівлі квитків на потяг (Рис. 1.6., 1.7.) або літак, банківські (Рис. 1.8.) або розважальні послуги, та багато чого іншого.

З прикладів бачимо, що сфери застосування у чат-ботів можуть дуже різними та охоплювати майже всі можливі випадки. Розповсюдженість такого виду взаємодії з користувачем зумовлена його зручністю, простоті та звичності. Адже якщо людина листується кожен день зі своїми знайомими та друзями, їй не повинно бути складно розібратися з таким форматом автоматизації. Зазвичай, розробники намагаються зробити алгоритм та інтерфейс взаємодії якомога зручнішим та інтуїтивно зрозумілішим для користувача. Функціонал, особливості та конкурентні переваги повністю визначаються автором застосунку.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Приклади ботів для бронювання та купівлі квитків на потяг у:

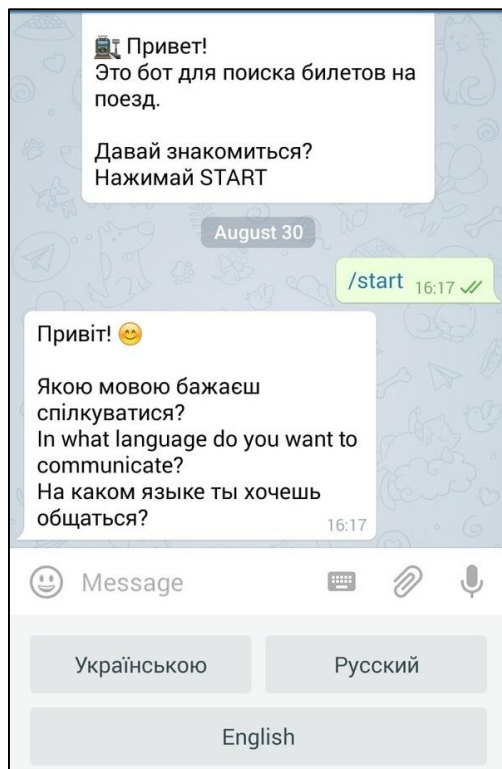


Рис. 1.6. Telegram.

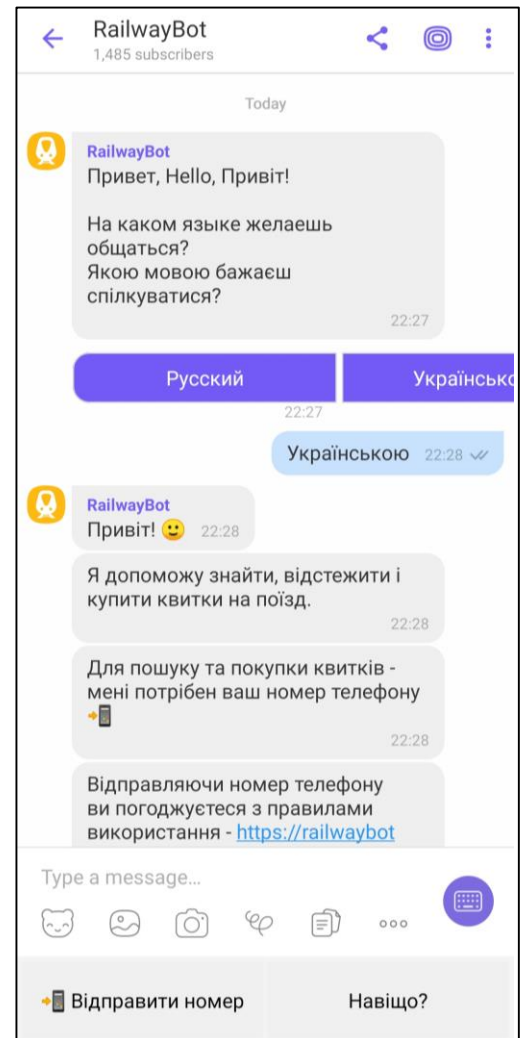


Рис. 1.7. Viber.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

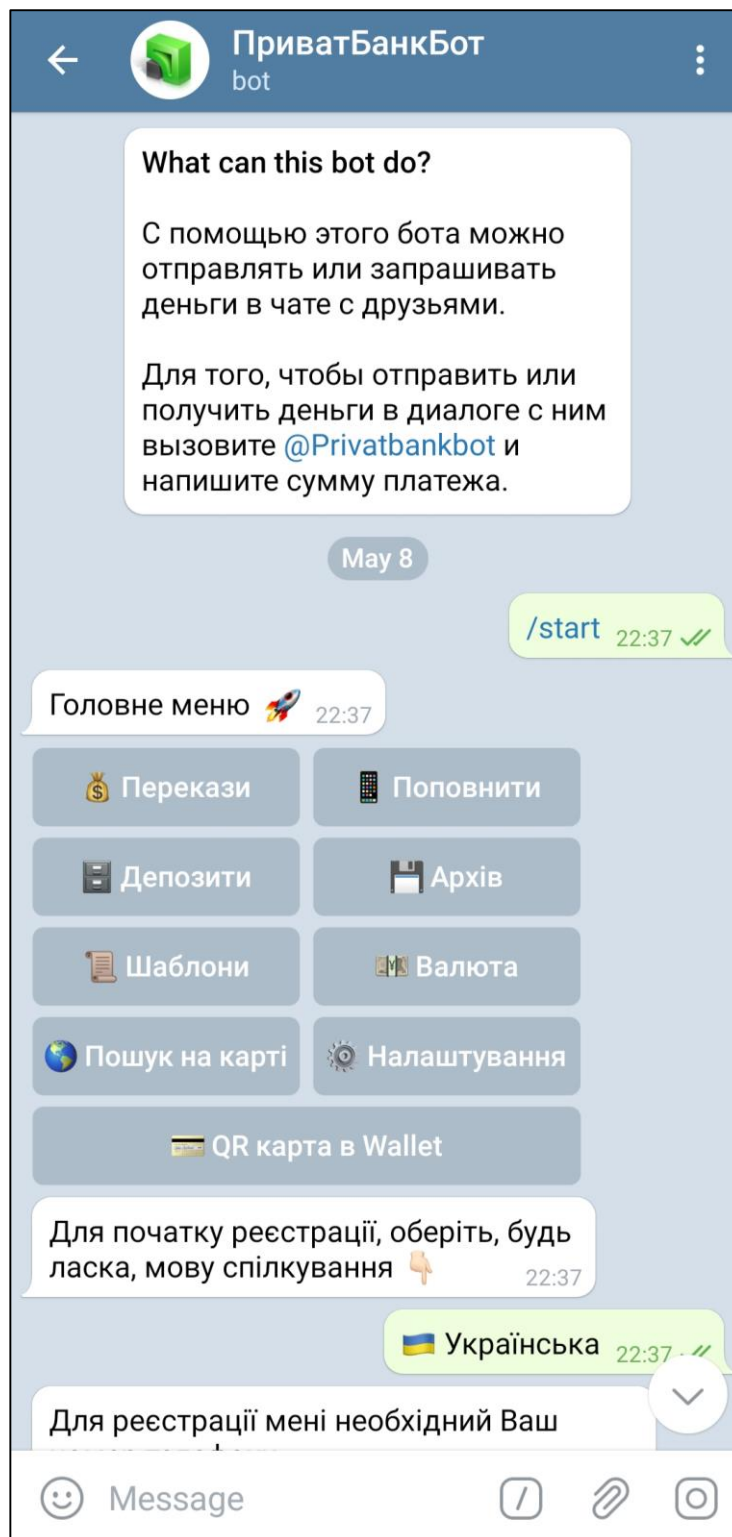


Рис. 1.8. Приклад боту у Telegram для банківських операцій.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Подібний стиль взаємодії не є новим. Він відомий відносно давно, з лише однією відмінністю – рутинну роботу раніше виконували люди.

Наприклад на різних сайтах іноді зустрічається спливаюче вікно чату з консультантом. У службах підтримки, кол-центрах, кінотеатрах, ресторанах та інших закладах обслуговування клієнтів завжди є певний набір рутинних операцій який можна було б автоматизувати.

Розглянувши основні переваги та недоліки виконання людиною подібних задач, виділимо наступні пункти.

Переваги:

- Індивідуальний, або максимально до нього наближений, підхід до кожної ситуації.
- Найпростіша для розуміння – «жива» мова спілкування.

Недоліки:

- Людський фактор може спричинити виникнення помилок.
- З огляду на різницю у кількості співробітників та клієнтів – можливі черги та затримки під час обслуговування.
- Можливі непередбачувані ситуації які обмежать кількість співробітників – хвороби, дикретні відпустки, нещасні випадки і т.д.
- Необхідність навчання нових співробітників необхідним комунікативним та професійним навичкам.
- Можливе неякісне виконання своїх обов’язків зі сторони співробітників.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

Проаналізувавши, перераховані вище переваги та недоліки залучення живого персоналу для виконання подібних задач, бачимо, що кількість недоліків та їх значущість іноді може перевищувати кількість переваг та їх значущість – що не є добре та може призвести до негативного впливу на загальну продуктивність підприємства. Як варіант для вирішення цих проблем було створене та залучене до виконання спеціалізоване програмне забезпечення.

Перед подібними кроками у сторону автоматизації, важливо завжди аналізувати доцільність заміни живого персоналу. Основними факторами, які необхідно врахувати, можуть бути наступні:

- Співвідношення наявної кількості співробітників, залучених до роботи з клієнтами у дистанційному режимі.
- Економія на заробітній платі після введення засобу автоматизації.
- Витрати на створення та підтримку устаткування.
- Специфіку підприємства, предметну область та потенційну можливість до автоматизації.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Огляд та аналіз основ і допоміжних засобів для створення боту

На сьогоднішній день, створити чат-бот на основі свого додатку пропонують декілька компаній, серед них Viber, Facebook та Telegram. Існує безліч звичайних та відео-інструкцій від сторонніх розробників та ентузіастів. У кожній з вищезгаданих компаній наявна достатньо детальна офіційна документація стосовно написання власного боту.

Розглядаючи та порівнюючи, з моєї суб'єктивної точки зору, перераховані вище додатки та їх актуальність станом на сьогодні – найкращим вибором для створення боту став Telegram, і ось чому:

- Швидкодія – телеграм є найшвидшим з протестованих додатків як на ПК так і на мобільних пристроях. Перевага у швидкості стосується завантаження додатку, надсилання та отримання повідомлень, синхронізації з серверами та багато чого іншого.
- Оптимізація – додаток чудово працює як на нових так і на старих версіях програмного забезпечення та відносно застарілому апаратному забезпеченні.
- Цільова аудиторія – так склалося, що телеграм частіше сприймається як молодіжний месенджер, частково – це впливає на його користувачів. Дійсно, зазвичай це будуть представники тієї вікової категорії, для якої будуть максимально звичною така концепція взаємодії.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

- Простота створення – бот у телеграмі може створити будь-хто за декілька хвилин. Для цього необхідно скористатися заздалегідь створеним ботом для створення ботів – тавтологія, та це вже гарний приклад використання такого роду автоматизації.
- Свобода для розробки – під цією голосною назвою зачаївся той факт, що створених ботів ніхто не контролює до тих пір, поки на них не почнуть надходити скарги, наприклад через авторські права або подібні. Відсутність модерації ботів дає змогу швидше створити та почати роботу боту.
- Широкі можливості – бот, створений на основі телеграму, має дуже великі можливості та широкий спектр сервісів для інтеграції.

Варто згадати і про недоліки перед конкурентами, найбільш очевидні – відсутність відео-зв'язку, та для наших цілей це не є проблемою.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4.1 Перші кроки до створення чат-боту на основі додатку Telegram

З офіційної документації бачимо, що для створення боту на основі додатку Telegram, нам необхідні:

- Попередня реєстрація у Telegram за номером мобільного номеру.
- Наявність програмного та апаратного забезпечення для написання та тестування.
- Стабільне підключення до мережі інтернет.

Наступною дією буде створення/реєстрація нашого нового боту. Для цього скористаємося спеціальним ботом (Рис. 1.9.).

					ІАЛЦ. 467800.003 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

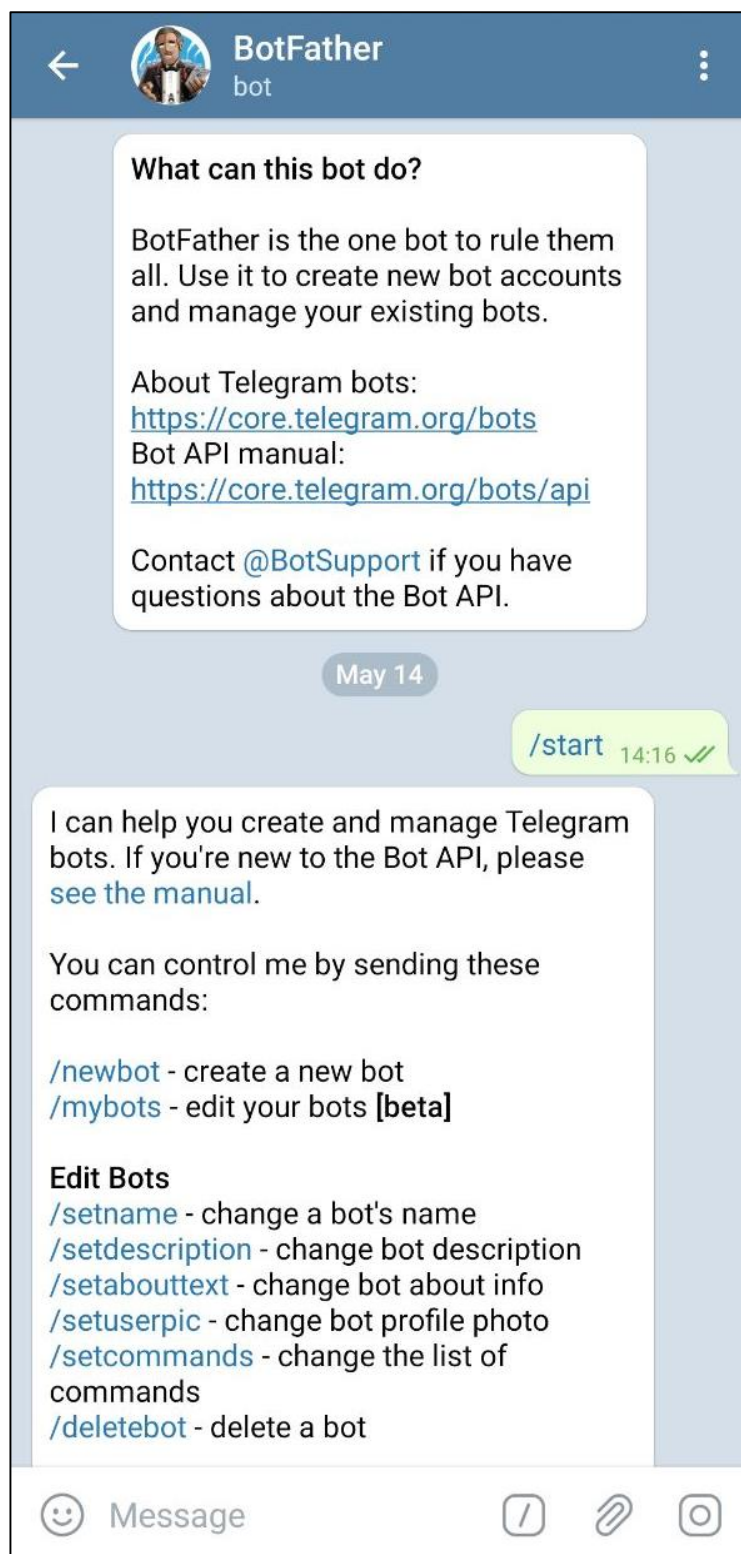


Рис. 1.9. Бот для створення ботів.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

У якості відповіді отримуємо перелік наступних команд:

/newbot – Створення нового боту.

/mybots – Редагування наявних, раніше створених ботів.

Редагування:

- /setname – Зміна імені.
- /setdescription - Зміна опису.
- /setabouttext - Зміна тексту стосовно можливостей.
- /setuserpic - Зміна зображення.
- /setcommands – Зміна списку команд.
- /deletebot – Видалення боту.

Налаштування:

- /token – Генерація ключу доступу.
- /revoke – Нова генерація ключу доступу.
- /setinline – Увімкнення режиму вбудованої клавіатури.
- /setinlinegeo – Увімкнення режиму вбудованої клавіатури для відслідковування геолокації.
- /setinlinefeedback – Зміна змісту вбудованої клавіатури.
- /setjoiningroups – Можливість участі боту у групових бесідах.
- /setprivacy – Налаштування приватності у групових бесідах.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

Ігри:

- /mygames – Редагування наявних, створених раніше ігор.
- /newgame – Створення нової гри.
- /listgames – Перелік створених нами раніше ігор.
- /editgame – Редагування гри.
- /deletegame – Видалення гри.

На перший погляд, така кількість текстових команд для керування може викликати відчуття, чимось схоже з панікою та відчаєм, та при більш уважному огляді бачимо, що кожна команда являє собою активне гіперпосилання, тобто відсутня необхідність ручного введення. Лаконічний опис кожної команди дає змогу одразу зрозуміти її призначення. До того ж, під час взаємодії з ботом – з’являються кнопки які дозволяють набагато легше виконувати ті чи інші операції.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

З цього різноманіття команд, скористаємося найпершою – /newbot. Далі необхідно обрати ім'я та юзернейм для нашого боту, ім'я буде у якості заголовку, юзернейм – для пошуку боту у базі наявних (Рис. 1.10.).

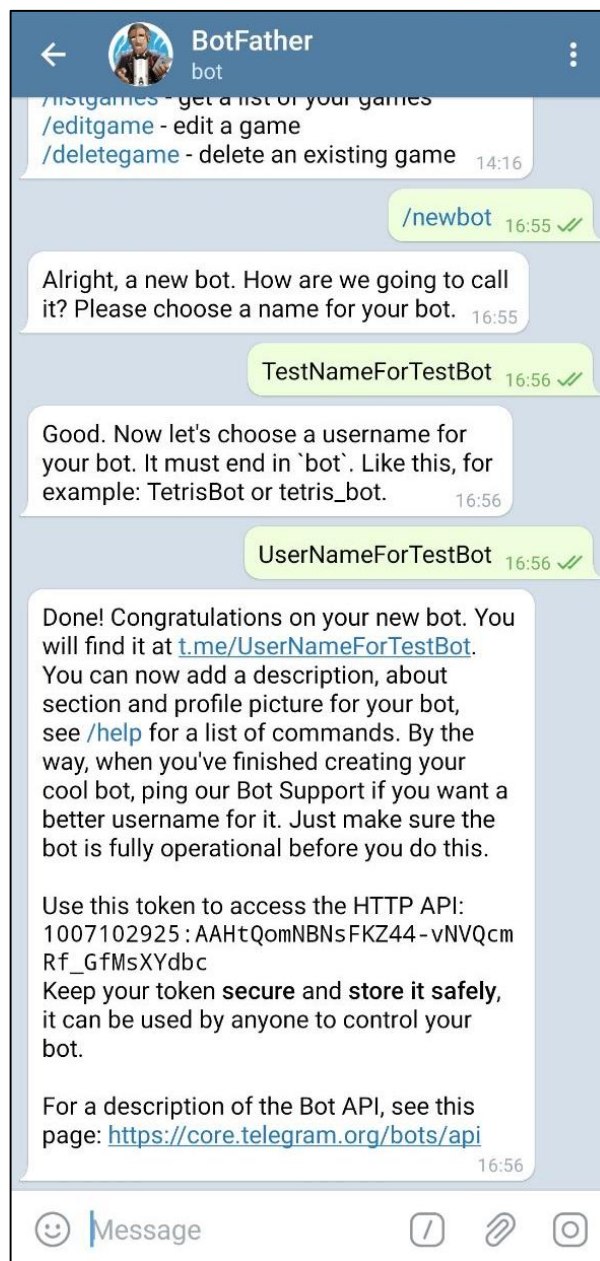


Рис. 1.10. Перші команди для створення боту.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо останні два етапи пройдено успішно – отримуємо посилання на бота та токен для подальшої програмної взаємодії. Бот створений (Рис. 1.11.), єдине що, він зараз не має жодного функціоналу, зображення, опису і т.д.

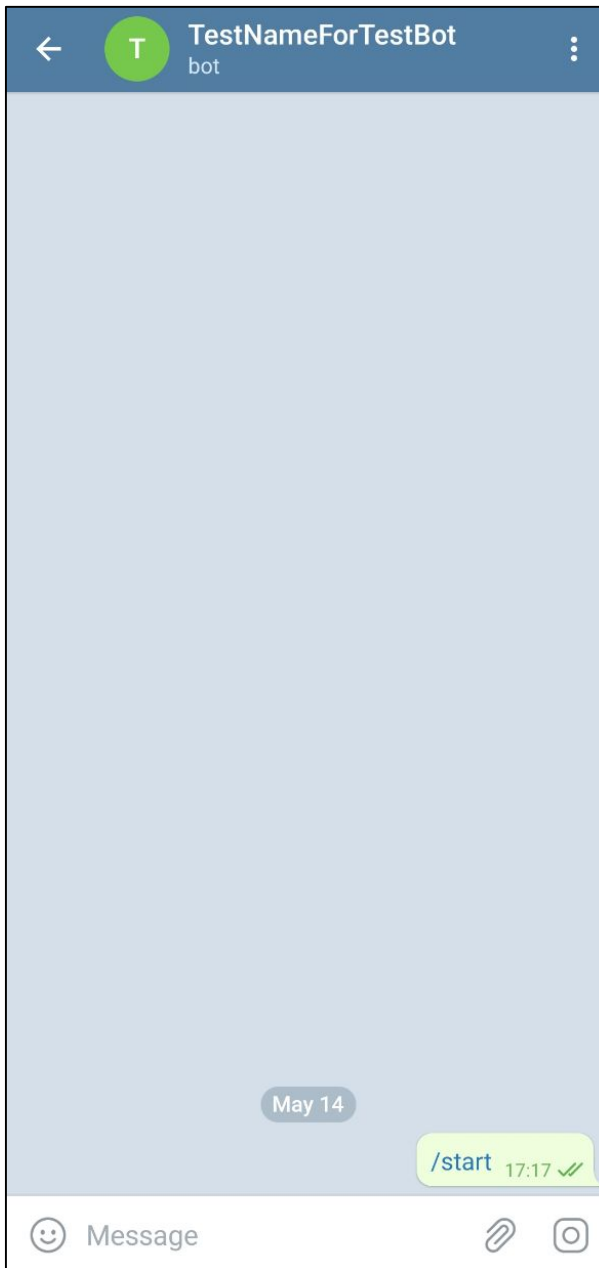


Рис. 1.11. Щойно створений бот.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Ввівши команду /mybots, та обравши щойностворений бот – отримуємо меню взаємодії для редагування (Рис. 1.12.).

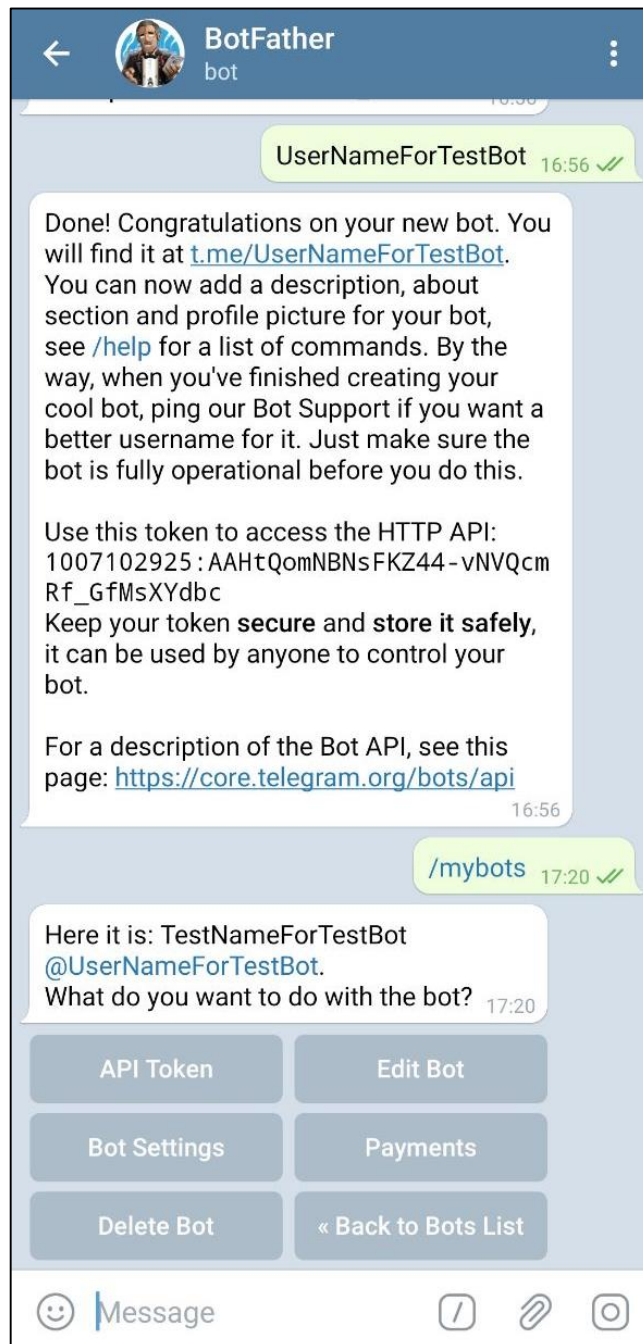


Рис. 1.12. Меню взаємодії для редагування.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 1

У підрозділі 1.1 був короткий екскурс у історію виникнення торгівлі як явища та змінення її форматів з плином часу.

У наступному підрозділі 1.2 були оглянуті основні види та способи ведення інтернет-торгівлі. Звернули увагу на переваги та недоліки кожного з них.

Підрозділ 1.3 розповідає про поняття "бот" та його відгалуження у вигляді чат-ботів, приклади їх реалізацій на різних основах та різноманітність сфер застосування.

У підрозділі 1.4 був створений власний чат-бот на основі додатку для обміну повідомленнями Телеграм для демонстрації та/або тестування. Оглянуті основні команди для керування вже створеними чат-ботами.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2.

ОГЛЯД ПРИЧИН, СЕРЕДОВИЩА, МОВ ПРОГРАМУВАННЯ, ДОПОМІЖНИХ БІБЛІОТЕК ТА ЗАСОБІВ ДЛЯ СТВОРЕННЯ І ПІДТРИМКИ РОБОТИ ЗАСТОСУНКУ

2.1 Аналіз необхідності створення власного боту

З попереднього пункту бачимо, що у сердніх та великих компаній є достатньо варіантів для того, щоб бути у тренді та полегшувати користування своїми послугами. Натомість, невеличким магазинам або щойно відкритим, зважаючи на обмежений бюджет, не залишається нічого, окрім використання соц. мереж для цих цілей.

Звісно, використання соц. мереж для розміщення каталогу свого магазину у мережі інтернет - не є чимось поганим або забороненим, та я вважаю що якщо вже звертатися до подібних рішень, то робити це максимально зручно, як для себе так і для клієнтів.

Тож метою даної дипломної роботи є створення кращого аналогу сторінкам у соціальних мережах, один із можливих способів це зробити – чат-бот.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

2.2 Вибір мови програмування

Написати чат бот для телеграму можна використовуючи одну з доступних мов програмування, серед них:

- Node.js
- PHP
- Python
- Java
- Ruby
- Swift

Це був перелік основних наявних мов програмування, повний список:

Node.js, PHP, Python, Java, Ruby, Swift, Go, Rust, Kotlin, C++, C#, Dart, Scala, Perl, Lua, Haskell, Ocaml. Для кожного з них, на офіційному сайті представлені приклади для різних бібліотек (Рис. 2.1).

Серед цього великого розмаїття мов програмування, був обраний Python версії 3.8+ і ось чому:

- Курс з даної мови програмування був у навчальній програмі університету, тож її використання дозволить продемонструвати корисність пройденого матеріалу.
- Функціональний стиль мови має багато спільного зі специфікою написання боту.
- Python – має простий синтаксис, та при цьому дуже потужний функціонал, також в ній відсутні певні обмеження, присутні у конкурентів.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

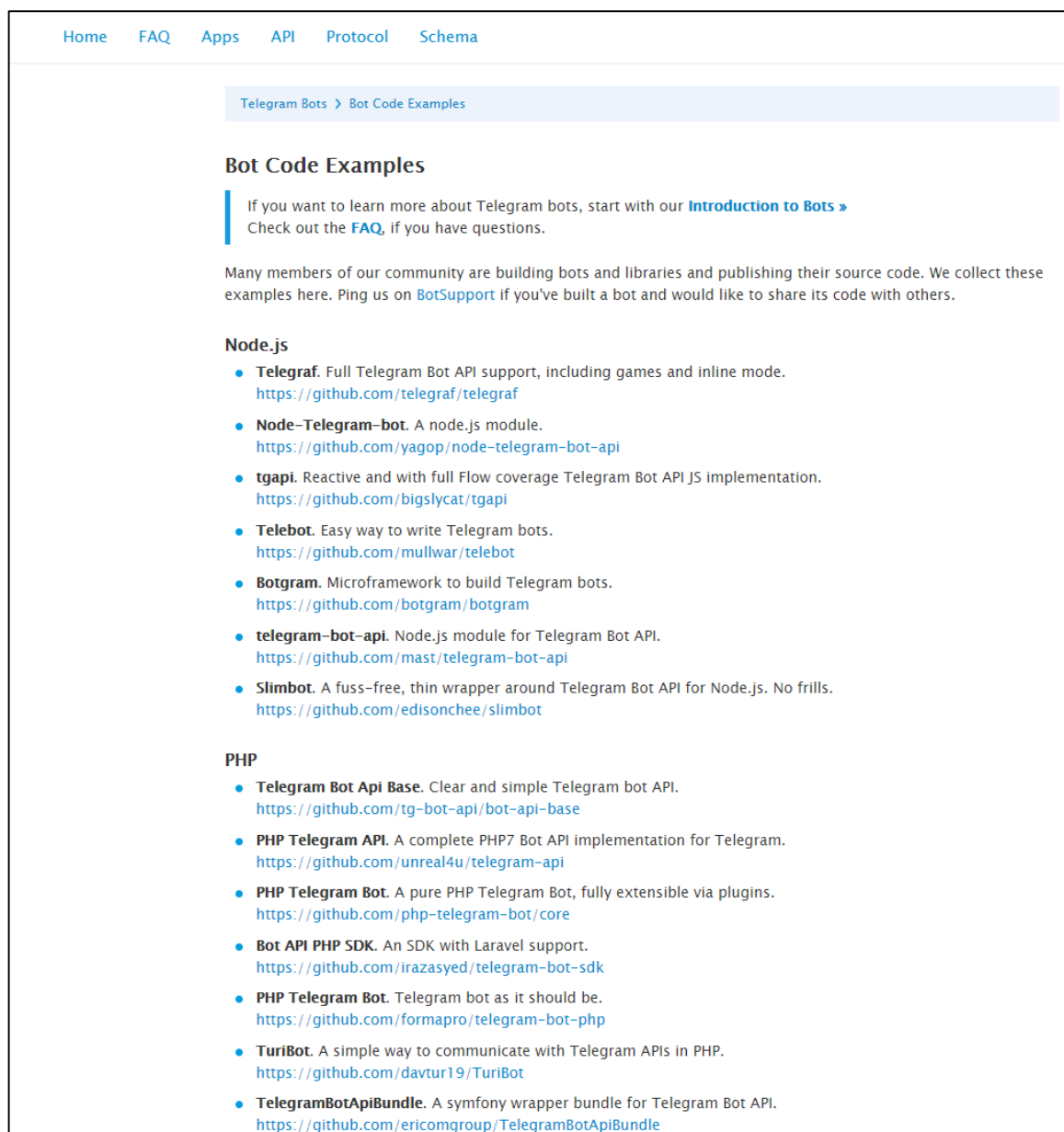


Рис. 2.1. Приклади з офіційного веб-сайту.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3 Аналіз допоміжних засобів для створення

Послуги, зі створення ботів на своїй основі, надають декілька додатків для обміну повідомленнями, серед них – «Telegram». Його конкурентні переваги були описані у попередньому У якості бази даних та місця зберігання зображень були обрані Google Sheets (Рис. 2.3.) та Google Photos (Рис. 2.2.).

Табличний процесор Google Sheets був обраний для цього проекту у якості бази даних з декількох причин:

- Можливість перегляду та редагування з практично будь-якого пристрою.
- Функціональні та зручні бібліотеки та інструменти для розробки програмного забезпечення.
- Звичний, простий та інтуїтивно зрозумілий інтерфейс.

Сервіс Google Photos дозволить мати у швидкому та зручному доступі фотографії товарів, для подальшого їх відправлення покупцю.

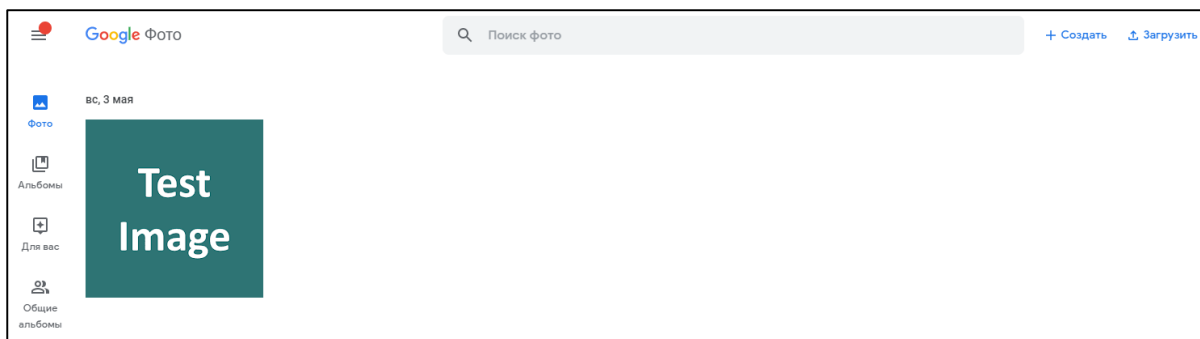


Рис. 2.2. Інтерфейс Google Photos.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

Diploma

Файл

Изменить

Вид

Вставка

Формат

Данные

Инструменты

Дополнения

Справка

Последнее изменение: electricity 2 дня назад

</

Рис. 2.3. Интерфейс Google Sheets.

					ИАЛЦ. 467800.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Огляд середовища розробки

Середовища розробки ні на що не впливають – таке переконання часто можна почути. Так, дійсно, код, написаний у «Блокноті» та у сучасному середовищі розробки нічим не відрізняються. Відмінність проявляється саме під час написання цього коду, адже розробниками редакторів коду передбачено безліч засобів спрощення, таких як перевірка на наявність помилок, швидкий запуск та тестування, автоматичне продовження, підсвічення синтаксису, простіше підключення сторонніх бібліотек, швидке форматування тексту та багато чого іншого.

Python – дуже поширена мова програмування, тож для неї існує безліч середовищ розробки від стандартного безкоштовного та вбудованого редактору до редакторів від сторонніх розробників, іноді платних.

Одним із найпоширеніших таких редакторів є PyCharm (Рис. 2.4.) – середовище розробки для мови Python від достатньо відомої компанії розробників JetBrains, у ньому і будемо працювати. Порівнюючи його зі стандартним IDLE (Рис. 2.5.) – можна виділити наступні переваги кожного редактору.

Переваги Pycharm:

- Засоби спрощення редагування коду, перераховані вище.
- Зручніший та більш функціональний інтерфейс, більше можливостей для його індивідуального налаштування.

Переваги стандартного IDLE:

- Займає меншу кількість пам'яті.
- Постачається разом із самою мовою програмування.
- Висока швидкодія під час запуску та інших стандартних операцій.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

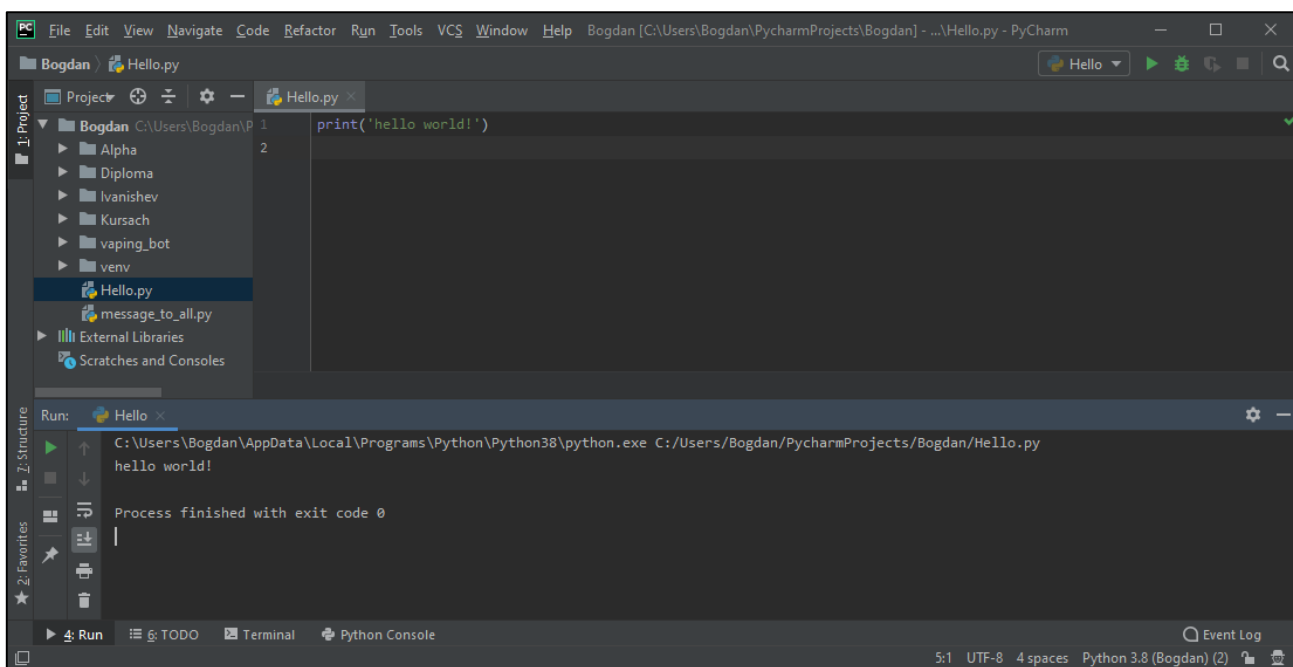


Рис. 2.4. Інтерфейс PyCharm.

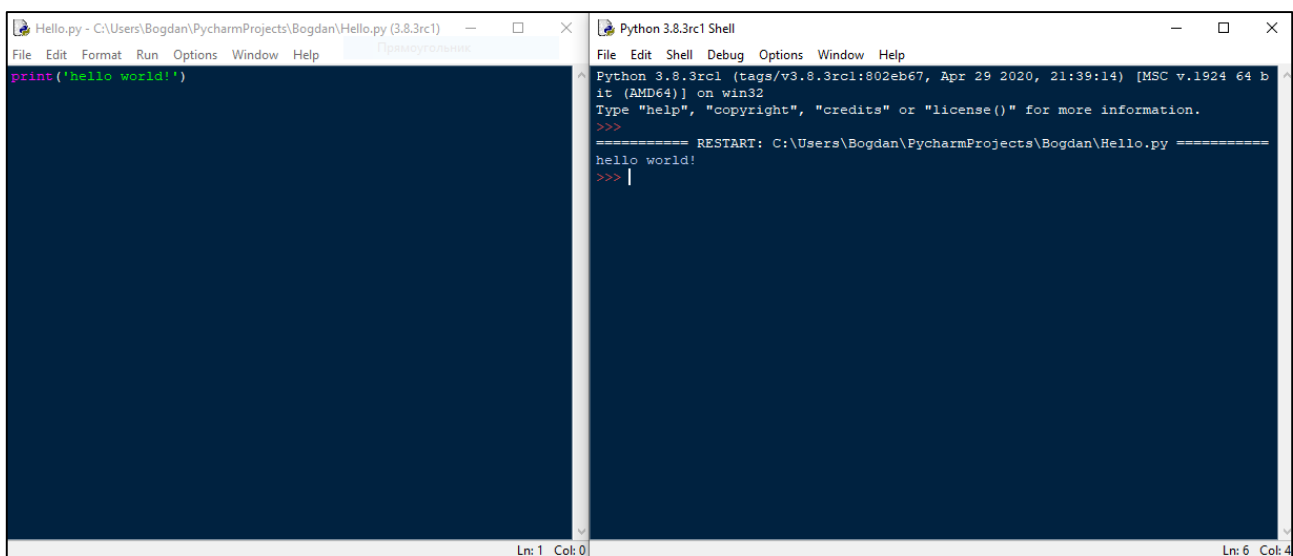


Рис. 2.5. Інтерфейс вбудованого IDLE.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

Отже, розробляти будемо бота, у середовищі PyCharm, на основі додатку для обміну повідомленнями Telegram, для обробки інтернет замовлень, цільовою аудиторією якого є представники малих магазини. Основними критеріями є простота використання, автоматизація вибору та оформлення замовлення, а також можливість його запуску та підтримки однією особою – власником.

Переваги бота на основі Телеграму у порівнянні з альтернативами:

- Автоматизація основних операцій.
- Телеграм – один із найпоширеніших месенджерів сьогодення, тож це позитивно вплине на простоту використання боту серед клієнтів.
- Розсилка новин клієнтам.
- Широкі можливості для підключення складної логіки застосунку.

Недоліки:

- Функціональні обмеження інтерфейсу.
- Майже повна відсутність налаштування дизайну.

Зваживши переваги та недоліки, було прийняте рішення про створення боту. Адже це дозволить ще більше спростити формат інтернет-магазину саме для невеликих магазинів та підприємців шляхом автоматизації основних кроків взаємодії з покупцем.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3.

НАПИСАННЯ ВЛАСНОГО БОТУ

3.1 Загальний огляд проекту

Трирівнева архітектура проекту (Рис. 3.1.). – це один із найбільш розповсюджених підходів до проектування додатку на сьогоднішній день. Існують певні варіації, та для дипломного проекту був обраний саме базовий варіант, що включає в себе рівні клієнту, логіки та даних. Провівши паралелі та конкретизувавши – отримаємо наступні рівні проекту:

1. Чат-бот на основі Телеграму – рівень клієнта.
2. Програмний код для обробки команд – рівень логіки.
3. База даних у Google Sheets та Google Photos – рівень даних.

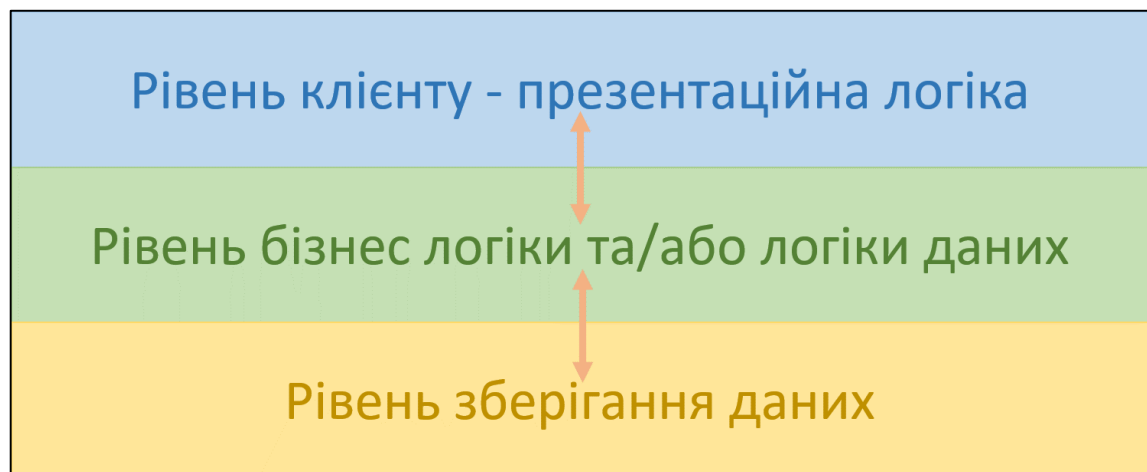


Рис. 3.1. Трирівнева архітектура додатку.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

3.1.1 Рівень клієнту

Як було сказано раніше, у якості рівня клієнту – виступає чат-бот на основі додатку для обміну повідомленнями Телеграм. Тому при огляді даного рівня буде продемонстровано псевдо-інтерфейс, реалізований за допомогою вбудованих цифрових кнопок для взаємодії з ботом. Щоб мати можливість визначитися з мінімальним набором функцій боту – розглянемо абстрактний магазин товарів для тварин з двома категоріями продукції.

Функціонал зі сторони покупця:

- Клієнт повинен мати змогу обрати товар серед представлених – реалізуємо це проходженням крізь низку фільтрів (Рис. 3.2.).
- Окрім самого товару необхідно також мати змогу вказати бажану кількість серед наявної (Рис. 3.3.).
- Перед підтвердженням замовлення також, бажана, можливість оглянути свою покупку – додаток повинен надіслати заготовлене зображення товару (Рис. 3.4.).
- Для кращої навігації додатком реалізуємо кнопку повернення на попередній крок.

Функціонал зі сторони власника:

- Додаток взаємодіє з Google Sheets для простого редагування списку товарів (Рис. 3.5.), моніторингу клієнтообігу та реєстрації замовлень (Рис. 3.6.).
- Google Photos – для зберігання зображень наших товарів.
- Перевагою буде сповіщення про нове замовлення (Рис. 3.7.).
- Можливість запуску додатку без стороннього втручання.

					ІАЛЦ. 467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

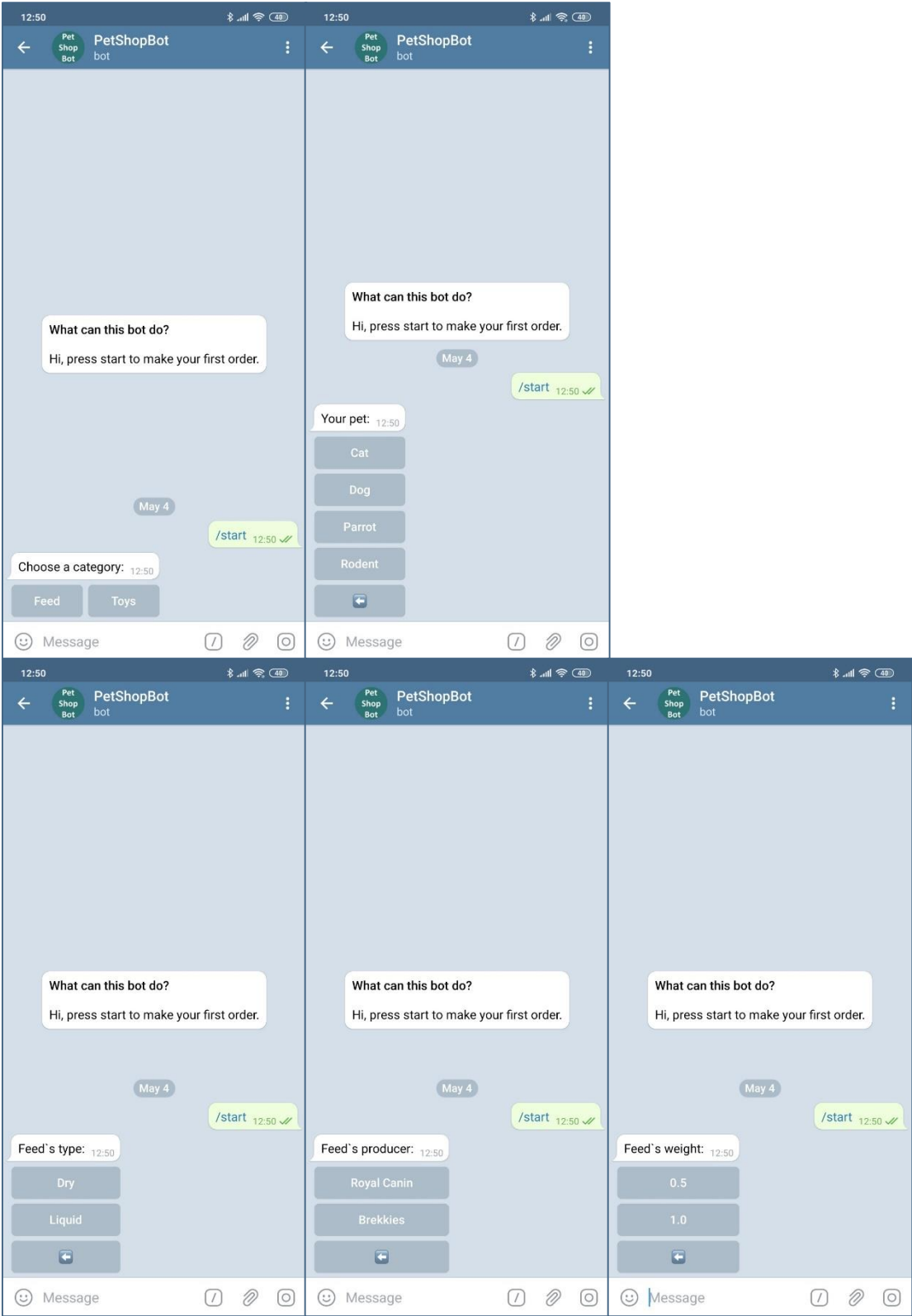


Рис. 3.2. Фільтрування товару.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

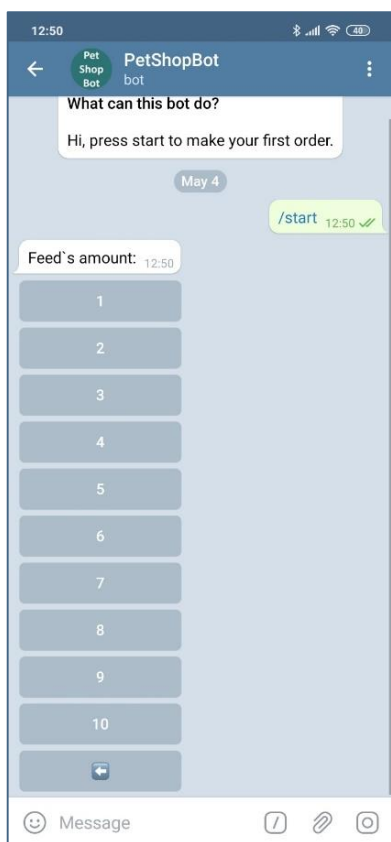


Рис. 3.3. Вибір кількості.

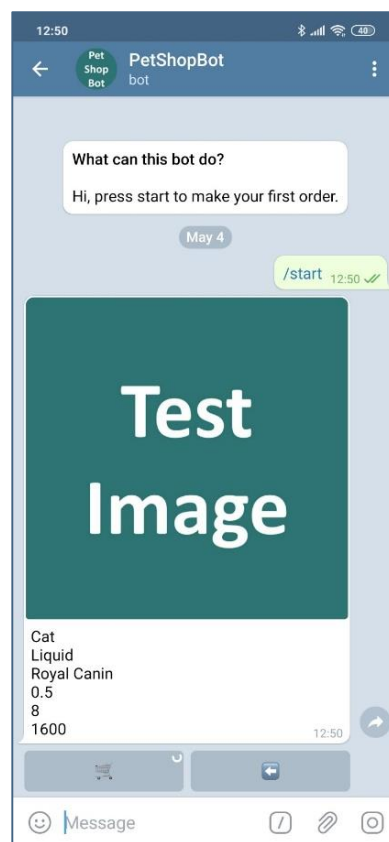


Рис. 3.4. Зображення товару.

A	B	C	D	E	F	G	H
f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Food							
Pet	Type	Producer	Weight	Amount	Price	Reserved	Picture
Cat	Dry	Royal Canin	0.5	10	200		https://lh3.googleusercontent.com/...
Cat	Dry	Royal Canin	1.0	10	350		https://lh3.googleusercontent.com/...
Cat	Dry	Brekkies	0.5	10	300		https://lh3.googleusercontent.com/...
Cat	Dry	Brekkies	1.0	10	550		https://lh3.googleusercontent.com/...
Cat	Liquid	Royal Canin	0.5	2	200	8	https://lh3.googleusercontent.com/...
Cat	Liquid	Royal Canin	1.0	10	400		https://lh3.googleusercontent.com/...
Cat	Liquid	Brekkies	0.5	10	100		https://lh3.googleusercontent.com/...
Cat	Liquid	Brekkies	1.0	10	180		https://lh3.googleusercontent.com/...
Cat	Liquid	Brekkies	2.0	10	180		https://lh3.googleusercontent.com/...

Рис. 3.5. Список товарів.

					ІАЛЦ. 467800.003 ПЗ		Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			40

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Customer							Order time	Orders						
User ID	Name	Second name	Username	Msg ID	Phone number	Registration date		Pet	Type	Producer	Weight Size	Amount	Total price	Promo
341219282	Bogdan	Kuzmenko	bodyanka38	12		Tue Mar 31 17:14:18 2020	Mon May 4 12:50:41 2020	Cat	Liquid	Royal Canin	0.5	8	1600	
							Sun May 3 11:54:50 2020	Dog	Dry	Royal Canin	0.5	4	800	
							Wed Apr 29 00:14:01 2020	Dog	Dry	Royal Canin	1.0	1	350	
							Wed Apr 29 00:13:19 2020	Dog	Soft	Trixie	Small	6	1200	
							Mon Apr 6 23:28:03 2020	Dog	Liquid	Royal Canin	0.5	2	400	
							Mon Apr 6 21:55:17 2020	Parrot	Dry	RIO	1.0	10	1300	
							Mon Apr 6 21:21:18 2020	Dog	Liquid	Brekkies	1.0	6	2100	
							Thu Apr 2 18:45:04 2020	Rodent	Dry	Vitapol	1.0	6	540	
							Thu Apr 2 18:30:56 2020	Cat	Dry	Royal Canin	1.0	6	2100	
							Tue Mar 31 17:15:06 2020	Dog	Dry	Royal Canin	1.0	4	1400	

Рис. 3.6. Моніторинг клієнтообігу та реєстрації замовлень.

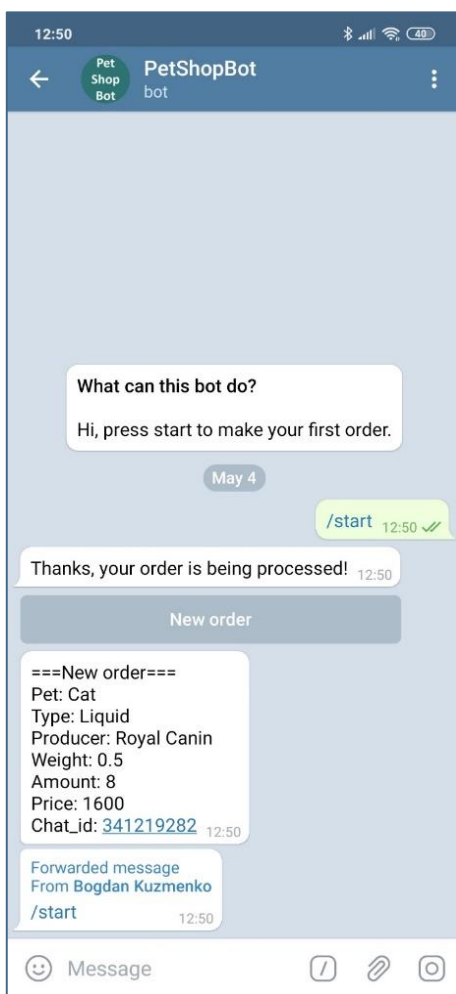


Рис. 3.7. Сповіднення про нове замовлення.

					ІАЛЦ. 467800.003 ПЗ		Арк.
							41
Зм.	Арк.	№ докум.	Підпис	Дата			

3.1.2 Рівень логіки

Логічна частина додатку буде реалізована на мові програмування Python версії 3, архітектура додатку (Рис. 3.8.) буде наступною:

- Main.py – основний виконуваний скрипт.
- Keyboards.py – скрипт для обробки натискань на цифрову клавіатуру зі сторони телеграму.
- Get_all.py – скрипт, відповідальний за зчитування статичної інформації з бази даних під час запуску програми.
- Functions.py – файл з усіма основними функціями, наприклад для початку роботи застосунку, реєстрації нового користувача, певної реакції на текст та пересування фільтрами.
- Feed.py, Toys.py – індивідуальні файли для кожної з категорій товарів.
- Client_secret.json – інформація для автентифікації додатку для взаємодії з сервісами Google.

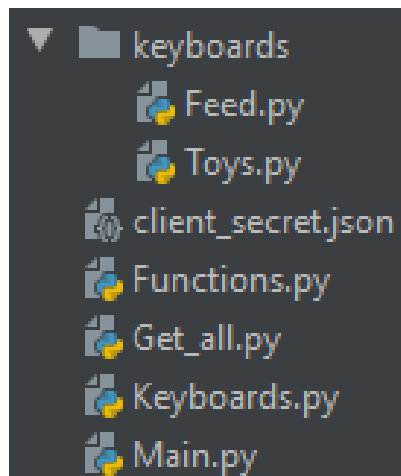


Рис. 3.8. Архітектура додатку.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Зі сторони користувача взаємодія із додатком виглядає наступним чином:

- Завантаження, встановлення та реєстрація у Telegram.
- Пошук боту серед представлених.
- Ознайомлення з функціоналом та початок роботи.
- Вибір та замовлення товару шляхом натискання клавiш віртуальної клавіатури.
- Після останнього кроку – очікування зв'язку зі сторони продавця для обговорення деталей доставки, оплати і т. д.

Взаємодія зі сторони власника:

- Завантаження, встановлення та реєстрація у Telegram.
- Налаштування таблиць зі змістом магазину.
- Налаштування робочого середовища на комп'ютері та запуск програми.
- Після сповіщення про нове замовлення - контакт з покупцем для обговорення деталей доставки, оплати і т. д.

Варто додати, що існує спосіб автоматизувати оплату карткою, та в нашому випадку, бажано, все ж уточнити замовлення перед оплатою. Окрім цього, автоматизація оплати має певні юридичні особливості, які для демонстрації було вирішено не виконувати.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Перша дія при запуску чат-бота – зчитати всю інформацію з бази даних, таку як поточна кількість наявних та зарезервованих товарів, посилання на їх зображення і т.д. Для універсального використання при різній кількості фільтрів – процес зчитування був уніфікований та реалізований у окремому файлі `Get_all.py`.

При початку взаємодії із користувачем – необхідно перевірити чи це перший його «візит» до магазину, для цього звіримо його ідентифікаційний номер з наявними, за допомогою функції `Functions.new_customer()`. Якщо користувач новий – записати основні дані стосовно нього у базу та запропонувати обрати категорію товару, перевіривши при цьому, чи не був використаний спеціальний код для надання знижки. Після вибору товару серед наявного – ознайомлення із його зображенням та основною інформацією. Наступний крок зі сторони користувача – підтвердження покупки, у цьому разі необхідно сповістити клієнта про успішне оформлення та запропонувати зробити інше замовлення. Бажано надати можливість клієнту повернутися на попередній крок, для цього додано кнопку «назад».

Після оформлення замовлення клієнтом, власник буде сповіщений про нього, далі він зможе зв'язатися із покупцем для підтвердження та оплати, відповідні записи з деталями замовлення з'являться у таблиці, буде зарезервована обрана кількість товару.

Зважаючи на особливість алгоритму фільтрування товару, можливість вибору його кількості доступна лише після того як стає відомий передостанній критерій. Відповідно, іноді, може виникнути ситуація, коли товару не буде в наявності, у такому разі, на жаль, користувач побачить перед собою лише кнопку повернення у попереднє меню (Рис. 3.9.) .

					ІАЛЦ. 467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

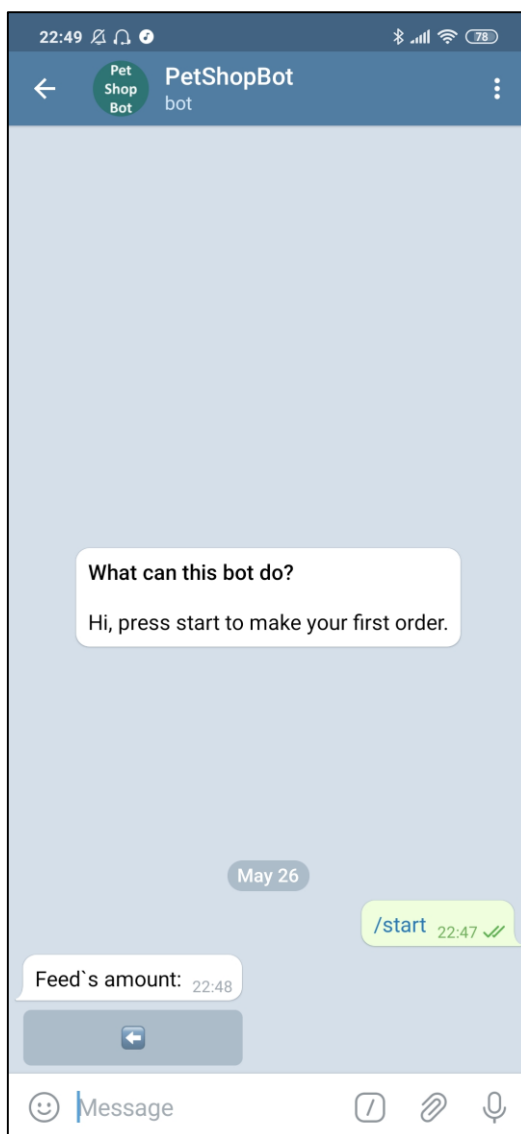


Рис. 3.9. Приклад результату роботи додатку у разі відсутності товару.

Для спрощення виконання були імпортовані наступні бібліотеки:

- Gspread – взаємодія та редагування таблиць Google Sheets за допомогою Python.
- Python-telegram-bot – взаємодія Telegram за допомогою Python.
- Oauth2client – бібліотека для автентифікації програмного забезпечення як такого, що має доступ до таблиць.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

3.1.3 База даних

Для взаємодії з сервісами Google – необхідно активувати певні API (Рис. 3.10.), а саме Google Drive API та Google Sheets API. Після реєстрації у якості розробника – отримуємо дані для авторизації програмного забезпечення для взаємодії з сервісами Google, зберігаються вони у файлі Client_secret.json.

Останній крок – надати доступ до таблиці для її редагування програмному забезпеченню (Рис. 3.11.)

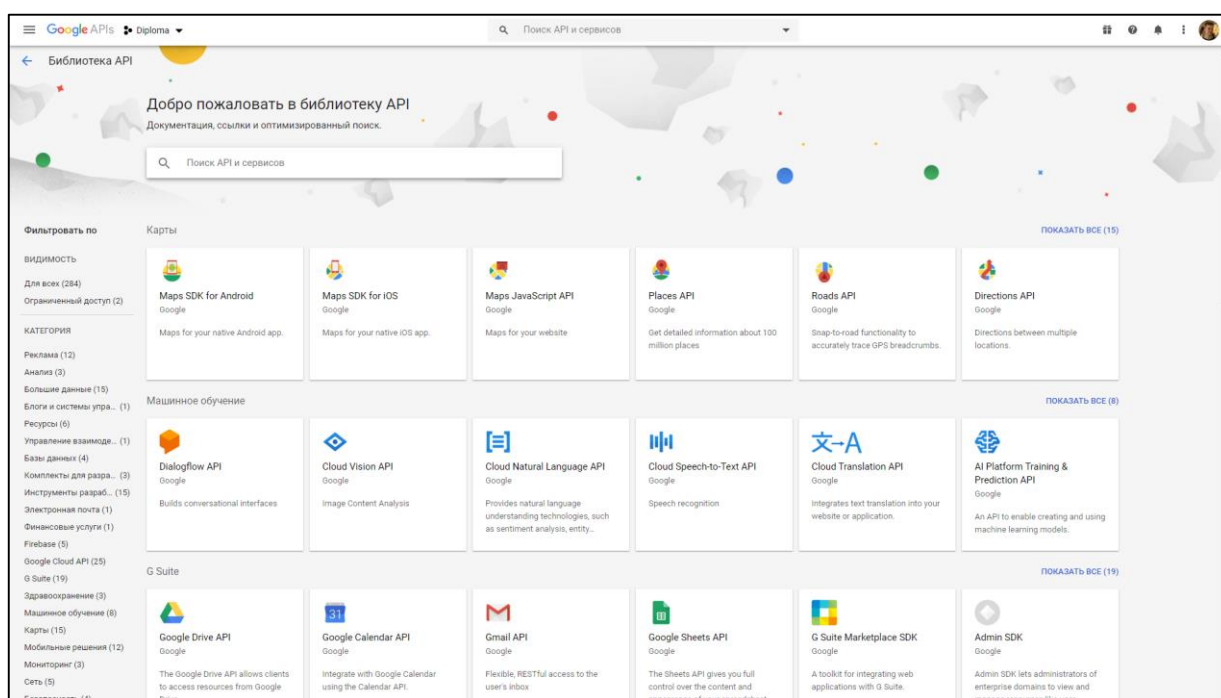


Рис. 3.10. Веб-сторінка для активації необхідних API.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

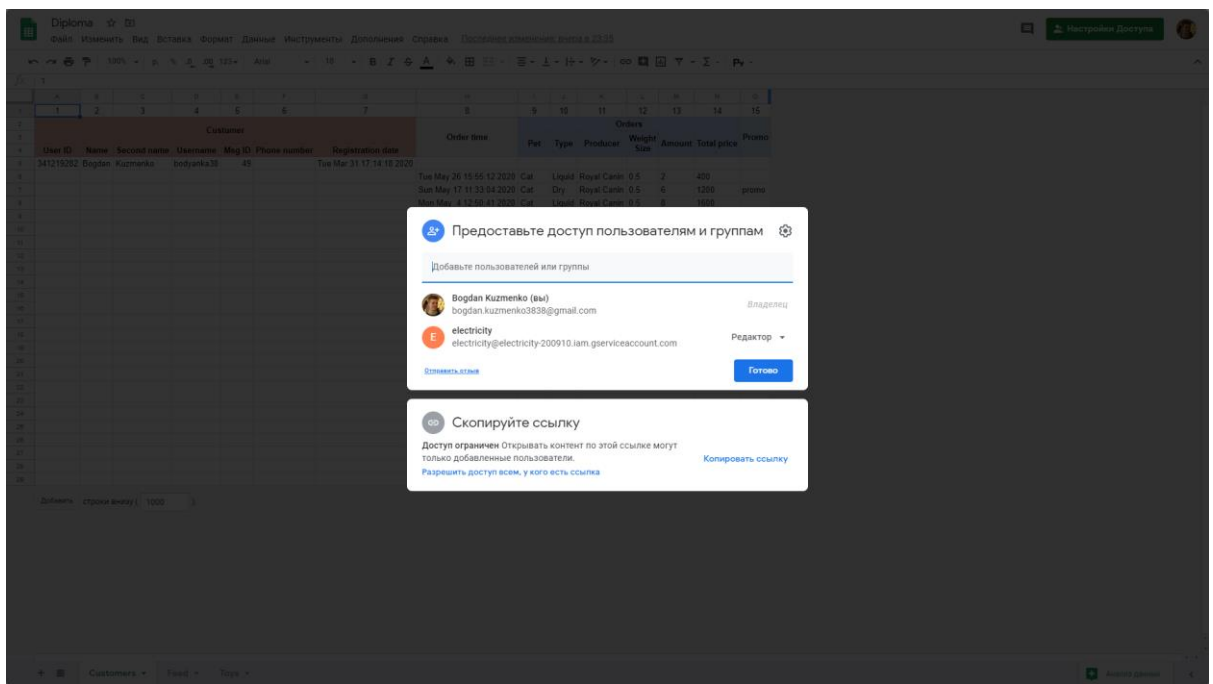


Рис. 3.11. Надання доступу до редагування.

Приклад організації даних у базі продемонстрований у попередньому підрозділі на рисунках 3.5. та 3.6. Таким чином, можна легко бачити історію усіх замовлень кожного клієнта та редагувати деталі товарів.

При появі нового клієнта – секція з його замовленнями буде додана у порожні рядки, що з’являться над вже існуючими.

Зображення товарів зберігаються у сервісі Google Photos в окремому альбомі (Рис. 2.2.). Такий підхід, знову ж таки, дозволяє просте додавання нових зображень практично з будь-якого пристрою шляхом копіювання посилання на фото, та вставлення у відповідне поле у базі даних.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 3

У цьому розділі був коротко описаний процес створення основних частин проєкту. Була написана логічна частина, зарєстрований та налаштований чат-бот у додатку Телеграм, створена БД. Продумана та реалізована взаємодія цих частин. Результатом є програма, яку можна запустити з ПК, при наявності необхідного ПО, чат-бот у додатку Телеграм та таблиця з даними. Програмний код взаємодії з чат-ботом та таблицями завдяки спеціальним стороннім бібліотекам.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час підготовки до створення проєкту було вивчено чимало теоретичного матеріалу, розглянуто декілька можливих варіантів реалізації. Необхідність звертатися до різноманітних документації та допоміжних інструкцій не зникла і під час реалізації додатку. Під час перших тестувань була виявлена особливість взаємодії із таблицями Google Sheets яка поставила існування проєкту, у його початковому вигляді, під питання. Після великої кількості спроб виправити цю прикру несподіванку, нарешті, був знайдений вихід.

Звісно, є певні моменти які потребують подальшого вдосконалення, та метою цього дипломного проєкту була реалізація базового функціоналу для демонстрації можливостей даного формату. На мою думку, це дуже цікавий та перспективний проєкт.

					ІАЛЦ. 467800.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Telegram bots – Режим доступу: <https://core.telegram.org/bots/samples> – Дата доступу: 29.05.2020.
2. GitHub – Режим доступу: <https://github.com/python-telegram-bot/python-telegram-bot> – Дата доступу: 29.05.2020.
3. Readthedocs – Режим доступу: <https://python-telegram-bot.readthedocs.io/en/stable/> – Дата доступу: 29.05.2020.
4. Facebook – Режим доступу: facebook.com – Дата доступу: 29.05.2020.
5. Instagram – Режим доступу: [Instagram.com](https://instagram.com) – Дата доступу: 29.05.2020.
6. Citrus – Режим доступу: citrus.ua – Дата доступу: 29.05.2020.
7. Google – Режим доступу: google.com – Дата доступу: 29.05.2020.

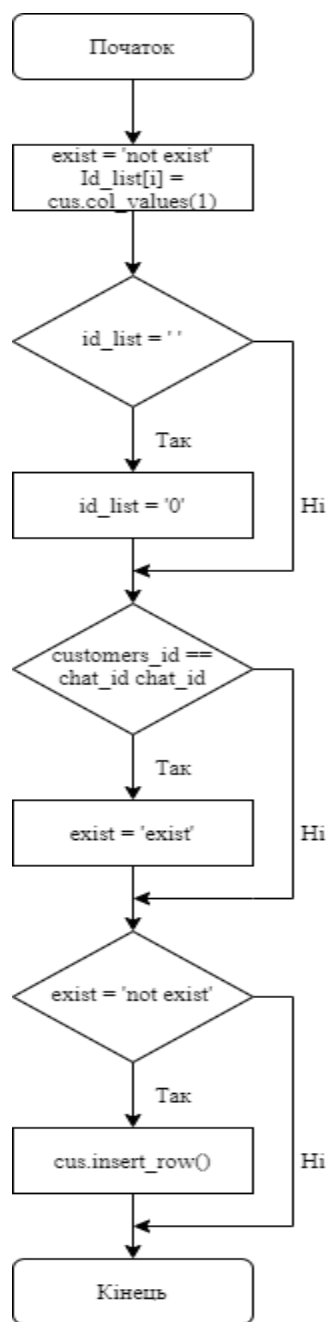
					ІАЛЦ. 467800.003 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А

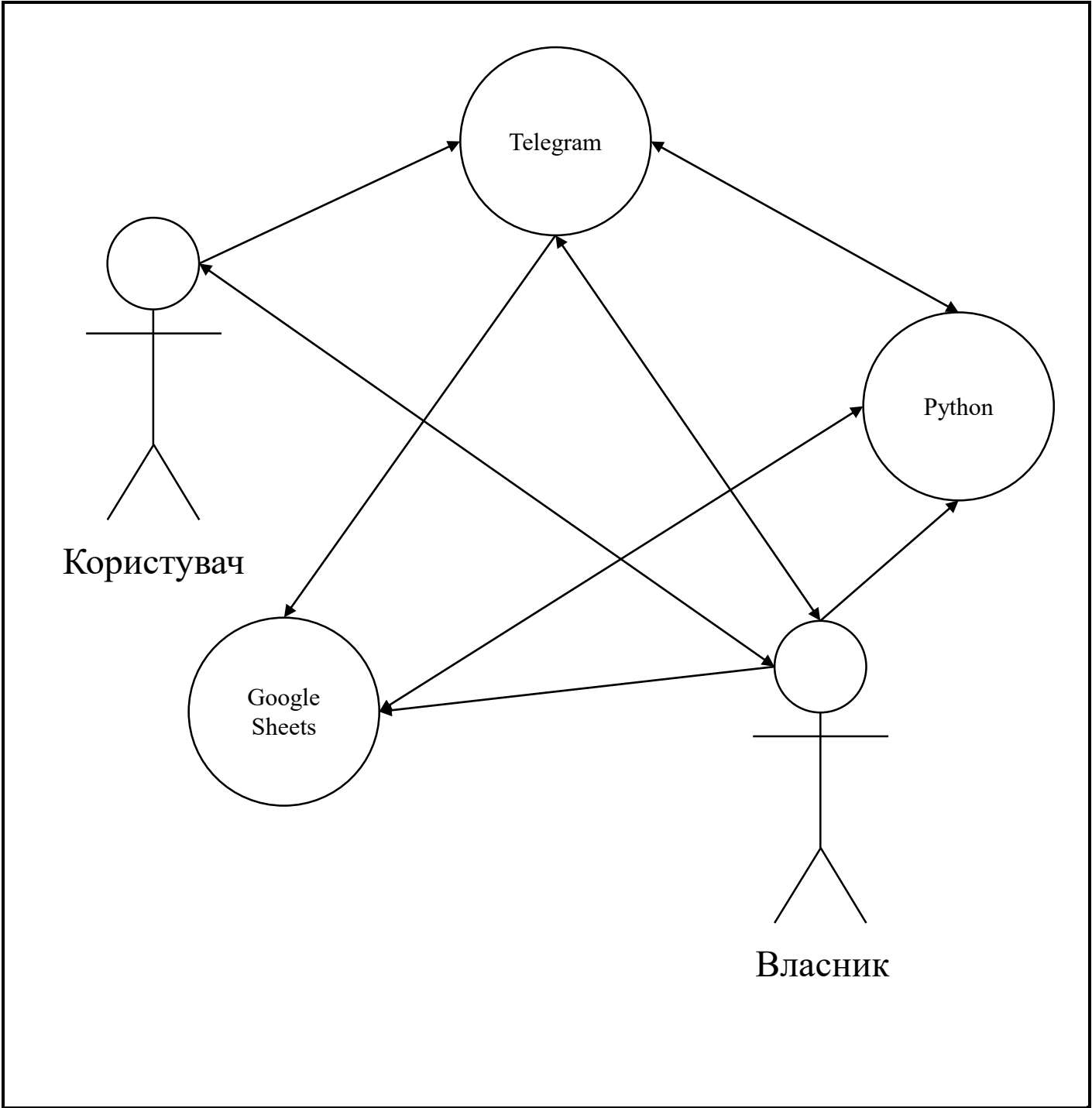
«Додаток для обробки інтернет замовлень»

КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ

Аркушів 3



					ІАЛЦ.467800.004 Д1									
					Додаток для обробки інтернет замовлень. Блок-схема алгоритму функції new_customer()									
Зм.	Аркуш	№ Докум.	Підпис	Дата										
Розробив		Кузьменко Б.С.												
Перевірів														
Т. Контр.						Аркуш 1		Аркушів 1						
					КПІ ім. Сікорського ФІОТ Група ІО-61									
Н. контр.		Сімоненко В.П.												
Затвердив														



					ІАЛЦ.467800.006 ДЗ					
					Додаток для обробки інтернет замовлень. Схема взаємодії проекту.					
Зм.	Аркуш	№ Докум.	Підпис	Дата						
	Розробив	Кузьменко Б.С.								
	Перевірив									
	Т. Контр.									
					КПІ ім. Сікорського ФІОТ Група ІО-61					
	Н. контр.	Сімоненко В.П.								
	Затвердив									
					Аркуш 1					
					Аркушів 1					

Додаток Б

«Додаток для обробки інтернет замовлень»

ЛІСТИНГ ПРОГРАМИ

Аркушів 14

Київ – 2020

Main.py

```
from telegram import *
from telegram.ext import *

import Diploma.Functions as ft
import Diploma.Keyboards as kb

from telegram.utils.request import *

logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO)
logger = logging.getLogger(__name__)

def error(update, context):
    logger.warning('Update "%s" caused error "%s"', update, context.error)

def main():
    request = Request(con_pool_size=20)
    bot = Bot(token='1161675658:AAEA3Tj1LaX-iqFVpP8IE3cstBRxWR5iKmE', request=request)
    updater = Updater(bot=bot)

    updater.dispatcher.add_handler(CommandHandler("start", ft.do_start))
    updater.dispatcher.add_handler(CommandHandler("help", ft.do_help))
    updater.dispatcher.add_handler(MessageHandler(Filters.text, ft.text))
    updater.dispatcher.add_handler(MessageHandler(Filters.contact, ft.contact_callback))

    updater.dispatcher.add_handler(CallbackQueryHandler(callback=kb.keyboards_callback_handler,
pass_chat_data=True))

    updater.dispatcher.add_error_handler(error)

    updater.start_polling(poll_interval=1)
    updater.idle()

if __name__ == '__main__':
    main()
```

					ІАЛЦ.467800.007 Д4					
					Додаток для обробки інтернет замовлень					
Зм.	Аркуш	№ Докум.	Підпис	Дата						
	Розробив	Кузьменко Б. С.								
	Перевірив	Регіда П. Г.								
	Т. Контр.									
					КПІ ім. Сікорського ФІОТ Група ІО-61					
	Н. контр.	Сімоненко В. П.								
	Затвердив	Регіда П. Г.								
					Аркуш 1		Аркушів 14			

Get_all.py

```
import gspread
from telegram import *
from oauth2client.service_account import ServiceAccountCredentials

creds = ServiceAccountCredentials.from_json_keyfile_name(
    'client_secret.json', ['https://www.googleapis.com/auth/drive'])
client = gspread.authorize(creds)

table = client.open('Diploma')

customers = table.worksheet("Customers").get_all_values()
feed = table.worksheet("Feed").get_all_values()
toys = table.worksheet("Toys").get_all_values()

def get_all_new(mass, source, num_of_filters):

    pos = 0
    result = []
    state = True
    a = [str(x) for x in range(1, int(get_max(source, num_of_filters-1)) + 1)]

    for i in source[4:]:
        for j in range(len(mass)):

            if i[j] == mass[j] or str(i[j]) in a:
                state = True
            else:
                state = False
                break

        if state and i[len(mass)] not in result and i[len(mass)] != "":
            result.append(i[len(mass)])
            pos = source.index(i)

    if len(mass) < num_of_filters:
        keyboard = []
        amount = source[pos][len(mass)]

        if len(mass) == num_of_filters - 1:
            for i in range(1, int(amount) + 1):
                keyboard.append([InlineKeyboardButton(i, callback_data=i)])
        else:
            for i in list(result):
                keyboard.append([InlineKeyboardButton(i, callback_data=i)])

    keyboard.append([InlineKeyboardButton('←', callback_data=str(source[0][len(mass)]))])
```

					ІАЛЦ. 467800.007 Д4	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    return mass, InlineKeyboardMarkup(keyboard)

elif len(mass) == num_of_filters:
    res_txt = "

    for i in mass:
        res_txt += i + '\n'

    res_txt += str(int(source[pos][num_of_filters])*int(mass[-1]))
    img = source[pos][num_of_filters + 2]
    price = source[pos][num_of_filters]
    return res_txt, img, price, pos

def get_max(source, num):
    temp_list = []
    for n in source[4:]:
        if n[num] != ":":
            temp_list.append(int(n[num]))
    return max(list(set(temp_list)))

def get_lists(source, num):
    temp_list = []
    for n in source[4:]:
        if n[num] != ":":
            temp_list.append(n[num])
    return list(set(temp_list))

f_pet = get_lists(feed, 0)
f_type = get_lists(feed, 1)
f_producer = get_lists(feed, 2)
f_weight = get_lists(feed, 3)

t_pet = get_lists(toys, 0)
t_type = get_lists(toys, 1)
t_producer = get_lists(toys, 2)
t_size = get_lists(toys, 3)

def category():
    keyboard = [[InlineKeyboardButton('Feed', callback_data="feed"),
                 InlineKeyboardButton('Toys', callback_data="toys")]]
    return InlineKeyboardMarkup(keyboard)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Functions.py

```
import time
import gspread
from telegram import *
import Diploma.Get_all as Ga
from oauth2client.service_account import ServiceAccountCredentials

admins = ['341219282']
promo_code = 'promo'

fd = 0
ty = 0

def new_customer(cus, chat_id, name, second_name, username, phone_number="",
adding_time=time.asctime()):
    exist = 'not exist'

    id_list = cus.col_values(1)

    for i in range(len(id_list)):
        if id_list[i] == "":
            id_list[i] = '0'

    for customers_id in id_list[4:]:
        if int(customers_id) == int(chat_id):
            print('User exist: ' + str(username))
            exist = 'exist'

    if exist == 'not exist': # если не входит - добавить нового покупателя
        print('New user: ' + str(username))
        cus.insert_row([chat_id, name, second_name, username, "", phone_number, adding_time], 5)

def do_start(bot: Bot, update: Update, optional='no'):
    global promo_code

    scope_n = ['https://www.googleapis.com/auth/drive']
    creds_n = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope_n)
    client_n = gspread.authorize(creds_n)

    customers_n = client_n.open('Diploma').sheet1

    new_customer(customers_n,
        chat_id=update['message']['chat']['id'],
        name=update['message']['chat']['first_name'],
        second_name=update['message']['chat']['last_name'],
        username=update['message']['chat']['username'],
```

					ІАЛЦ. 467800.007 Д4	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        adding_time=time.asctime())

chat_id = update['message']['chat']['id']
msg_id = update['message']['message_id']

id_list = customers_n.col_values(1)

for i in range(len(id_list)):
    if id_list[i] == str(chat_id):
        customers_n.update_cell(i + 1, 5, msg_id)

if optional == promo_code:
    for j in range(len(id_list)):
        if id_list[j] == str(chat_id):
            if customers_n.cell(j + 1, 15).value != promo_code:
                bot.send_message(chat_id=chat_id, text='Promo code activated!')
                customers_n.update_cell(j + 1, 15, optional)
            else:
                bot.send_message(chat_id=chat_id, text='This promo code has already been activated!')

bot.send_message(chat_id=chat_id, text="Choose a category:", reply_markup=Ga.category())

def contact_callback(bot: Bot, update: Update):
    scope_n = ['https://www.googleapis.com/auth/drive']
    creds_n = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope_n)
    client_n = gspread.authorize(creds_n)

    customers_n = client_n.open('Customers_vol2').sheet1

    contact = update.effective_message.contact
    phone = contact.phone_number

    chat_id = update['message']['chat']['id']
    id_list = customers_n.col_values(1)

    for i in range(len(id_list)):
        if id_list[i] == str(chat_id):
            customers_n.update_cell(i + 1, 6, phone)

def text(bot: Bot, update: Update):
    global promo_code

    idi = str(update['message']['chat']['id'])
    texto = update['message']['text']
    mytext = texto.split("|")

    print(texto)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

```

if mytext[0] == 'send' and idi in admins:
    bot.send_message(chat_id=mytext[1], text=str(mytext[2][1:]))
    print(mytext)

elif mytext[0] == promo_code:
    do_start(bot, update, promo_code)

else:
    do_start(bot, update)

def do_help(bot: Bot, update: Update):
    texto = '/start - start of product selection\n'
    texto += '← - previous menu\n'
    texto += '👉 - order approvment'

    bot.send_message(chat_id=update.message.chat_id, text=texto)

def forward(q, d, b, mass, num_of_fil, dict_my, list_my, num, tp="", pp=""):

    temp = []
    for j in dict_my.values():
        temp.append(j)

    x = [temp[i] for i in range(num-1)]
    x.append(d)

    # first()
    if num == 0:
        x.pop()

    if num != num_of_fil:
        q.edit_message_text(text=list_my[num], reply_markup=Ga.get_all_new(x, mass, num_of_fil)[1])

    # обновление массива order_feed
    if num != 0:
        dict_my[num-1] = d

    if num == num_of_fil:
        b.deleteMessage(chat_id=q['message']['chat']['id'], message_id=q['message']['message_id'])

        b.send_photo(chat_id=q['message']['chat']['id'],
                    photo=Ga.get_all_new(x, mass, num_of_fil)[1],

                    caption=Ga.get_all_new(x, mass, num_of_fil)[0],

                    reply_markup=InlineKeyboardMarkup([
                        [InlineKeyboardButton('👉', callback_data=tp),

```

					ІАЛЦ. 467800.007 Д4	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        InlineKeyboardButton('←', callback_data=pp)]
    )
)

dict_my[num-1] = d
dict_my[num] = Ga.get_all_new(x, mass, num_of_fil)[2]

def back(q, b, mass, num_of_fil, dict_my, list_my, num):

    temp = []
    for j in dict_my.values():
        temp.append(j)

    x = [temp[i] for i in range(num)]

    if num == -1:
        q.edit_message_text(text='Choose a category:', reply_markup=Ga.category())

    elif -1 < num < num_of_fil - 1:
        q.edit_message_text(text=list_my[num], reply_markup=Ga.get_all_new(x, mass, num_of_fil)[1])

    else:
        b.deleteMessage(chat_id=q['message']['chat']['id'], message_id=q['message']['message_id'])
        b.send_message(chat_id=q['message']['chat']['id'],
            text=list_my[num],
            reply_markup=Ga.get_all_new(x, mass, num_of_fil)[1])

```

					ІАЛЦ. 467800.007 Д4	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

Keyboards.py

```
import time
from telegram import *

import Diploma.Get_all as ga
import Diploma.keyboards.Feed as kbf
import Diploma.keyboards.Toys as kbt

# оброботчик всех кнопок со всех клавиатур
def keyboards_callback_handler(bot: Bot, update: Update, chat_data=None, **kwargs):
    query = update.callback_query
    data = query.data

    kbf.feed(query, data, bot,
             query['message']['chat']['id'],
             [kbf.order_feed[0], kbf.order_feed[1], kbf.order_feed[2],
              kbf.order_feed[3], kbf.order_feed[4], kbf.order_feed[5]],
             time.asctime())

    kbt.toys(query, data, bot,
             query['message']['chat']['id'],
             [kbt.order_toys[0], kbt.order_toys[1], kbt.order_toys[2],
              kbt.order_toys[3], kbt.order_toys[4], kbt.order_toys[5]],
             time.asctime())

    # print('query from')
    # print(query['message']['message_id'])

    if data == 'New order':
        query.edit_message_text(text='Choose a category:',
                                reply_markup=ga.category())
```

					ІАЛЦ. 467800.007 Д4	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

Feed.py

```
import gspread
from telegram import *

import Diploma.Get_all as Ga
import Diploma.Functions as Kb_func

from oauth2client.service_account import ServiceAccountCredentials

order_feed = {0: "", 1: "", 2: "", 3: "", 4: "", 5: ""}
dict_list = ['Your pet:', 'Feed`s type:', 'Feed`s producer:', 'Feed`s weight:', 'Feed`s amount:']
feed_num_of_filters = 5

# mass_o = [pet_o, type_o, producer_o, weight_o, amount_o, price_o]

def feed_order(q, b, chat_id, mass_o, adding_time):
    creds_n = ServiceAccountCredentials.from_json_keyfile_name(
        'client_secret.json', ['https://www.googleapis.com/auth/drive'])
    client_n = gspread.authorize(creds_n)

    customers_n = client_n.open('Diploma').worksheet("Customers")
    feed_n = client_n.open('Diploma').worksheet("Feed")

    # getting line of order
    pos = Ga.get_all_new([mass_o[0], mass_o[1], mass_o[2], mass_o[3], mass_o[4]], Ga.feed,
        feed_num_of_filters)[3]

    cur_am = feed_n.cell(pos + 1, 5).value
    cur_res = feed_n.cell(pos + 1, 7).value
    if cur_res == "":
        cur_res = 0

    feed_n.update_cell(pos + 1, 5, int(cur_am) - int(mass_o[4]))
    feed_n.update_cell(pos + 1, 7, int(cur_res) + int(mass_o[4]))

    id_list = customers_n.col_values(1)
    msg_id = 0

    for i in range(len(id_list)):
        if id_list[i] == str(chat_id):
            msg_id = customers_n.cell(i + 1, 5).value

            customers_n.insert_row(
                ['', "", "", "", "", "", adding_time,
                    mass_o[0], mass_o[1], mass_o[2], mass_o[3], mass_o[4], int(mass_o[5]) * int(mass_o[4])], i + 2)

    b.deleteMessage(chat_id=q['message']['chat']['id'], message_id=q['message']['message_id'])
```

					ІАЛЦ. 467800.007 Д4	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

```

b.send_message(chat_id=q['message']['chat']['id'],
               text='Thanks, your order is being processed!',
               reply_markup=InlineKeyboardMarkup(
                   [[InlineKeyboardButton('New order', callback_data='New order')]]))

```

```

b.send_message(chat_id=341219282,
               text='===New order===\n' +
                   'Pet: ' + mass_o[0] + '\n' +
                   'Type: ' + mass_o[1] + '\n' +
                   'Producer: ' + mass_o[2] + '\n' +
                   'Weight: ' + mass_o[3] + '\n' +
                   'Amount: ' + mass_o[4] + '\n' +
                   'Price: ' + str(int(mass_o[5]) * int(mass_o[4])) + '\n' +
                   'Chat_id: ' + str(chat_id))

```

```

b.forward_message(chat_id=341219282,
                  from_chat_id=chat_id,
                  message_id=msg_id)

```

```

Ga.feed = client_n.open('Diploma').worksheet("Feed").get_all_values()

```

```

def feed(q, d, b, chat_id, mass_o, adding_time):
    if d == "feed":
        Kb_func.ty = 0
        Kb_func.fd = 1
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 0)

    elif d in Ga.f_pet and Kb_func.fd == 1:
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 1)

    elif d in Ga.f_type and Kb_func.fd == 1:
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 2)

    elif d in Ga.f_producer and Kb_func.fd == 1:
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 3)

    elif d in Ga.f_weight and Kb_func.fd == 1:
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 4)

    elif d in [str(x) for x in range(1, Ga.get_max(Ga.feed, 4) + 1)] and Kb_func.fd == 1:
        Kb_func.forward(q, d, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 5, 'add_feed', 'f_6')

    elif d == 'add_feed' and Kb_func.fd == 1:
        feed_order(q, b, chat_id, mass_o, adding_time)
        Kb_func.fd = 0

    elif d == 'f_1':
        Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, -1)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

```

elif d == 'f_2':
    Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 0)

elif d == 'f_3':
    Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 1)

elif d == 'f_4':
    Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 2)

elif d == 'f_5':
    Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 3)

elif d == 'f_6':
    Kb_func.back(q, b, Ga.feed, feed_num_of_filters, order_feed, dict_list, 4)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Toys.py

```
import gspread
from telegram import *

import Diploma.Get_all as Ga
import Diploma.Functions as Kb_func

from oauth2client.service_account import ServiceAccountCredentials

order_toys = {0: ", 1: ", 2: ", 3: ", 4: ", 5: "}
dict_list = ['Your pet:', 'Toy`s type:', 'Toy`s producer:', 'Toy`s size:', 'Toy`s amount:']
toys_num_of_filters = 5

# mass_o = [pet_o, type_o, producer_o, size_o, amount_o, price_o]

def toys_order(q, b, chat_id, mass_o, adding_time):
    creds_n = ServiceAccountCredentials.from_json_keyfile_name(
        'client_secret.json', ['https://www.googleapis.com/auth/drive'])
    client_n = gspread.authorize(creds_n)

    customers_n = client_n.open('Diploma').worksheet("Customers")
    toys_n = client_n.open('Diploma').worksheet("Toys")

    # getting line of order
    pos = Ga.get_all_new([mass_o[0], mass_o[1], mass_o[2], mass_o[3], mass_o[4]], Ga.toys,
        toys_num_of_filters)[3]

    cur_am = toys_n.cell(pos + 1, 5).value
    cur_res = toys_n.cell(pos + 1, 7).value
    if cur_res == "":
        cur_res = 0

    toys_n.update_cell(pos + 1, 5, int(cur_am) - int(mass_o[4]))
    toys_n.update_cell(pos + 1, 7, int(cur_res) + int(mass_o[4]))

    id_list = customers_n.col_values(1)
    msg_id = 0

    for i in range(len(id_list)):
        if id_list[i] == str(chat_id):
            msg_id = customers_n.cell(i + 1, 5).value

            customers_n.insert_row(
                ["", "", "", "", "", "", adding_time,
                    mass_o[0], mass_o[1], mass_o[2], mass_o[3], mass_o[4], int(mass_o[5]) * int(mass_o[4])], i + 2)

    b.deleteMessage(chat_id=q['message']['chat']['id'], message_id=q['message']['message_id'])
```

					ІАЛЦ. 467800.007 Д4	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

```

b.send_message(chat_id=q['message']['chat']['id'],
                text='Thanks, your order is being processed!',
                reply_markup=InlineKeyboardMarkup(
                    [[InlineKeyboardButton('New order', callback_data='New order')]]))

```

```

b.send_message(chat_id=341219282,
                text='===New order===\n' +
                    'Pet: ' + mass_o[0] + '\n' +
                    'Type: ' + mass_o[1] + '\n' +
                    'Producer: ' + mass_o[2] + '\n' +
                    'Size: ' + mass_o[3] + '\n' +
                    'Amount: ' + mass_o[4] + '\n' +
                    'Price: ' + str(int(mass_o[5]) * int(mass_o[4])) + '\n' +
                    'Chat_id: ' + str(chat_id))

```

```

b.forward_message(chat_id=341219282,
                  from_chat_id=chat_id,
                  message_id=msg_id)

```

```

Ga.toys = client_n.open('Diploma').worksheet("Toys").get_all_values()

```

```

def toys(q, d, b, chat_id, mass_o, adding_time):
    if d == "toys":
        Kb_func.ty = 1
        Kb_func.fd = 0
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 0)

    elif d in Ga.t_pet and Kb_func.ty == 1:
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 1)

    elif d in Ga.t_type and Kb_func.ty == 1:
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 2)

    elif d in Ga.t_producer and Kb_func.ty == 1:
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 3)

    elif d in Ga.t_size and Kb_func.ty == 1:
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 4)

    elif d in [str(x) for x in range(1, Ga.get_max(Ga.toys, 4) + 1)] and Kb_func.ty == 1:
        Kb_func.forward(q, d, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 5, 'add_toys', 't_6')

    elif d == 'add_toys' and Kb_func.ty == 1:
        toys_order(q, b, chat_id, mass_o, adding_time)
        Kb_func.ty = 0

    elif d == 't_1':
        Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, -1)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

```

elif d == 't_2':
    Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 0)

elif d == 't_3':
    Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 1)

elif d == 't_4':
    Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 2)

elif d == 't_5':
    Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 3)

elif d == 't_6':
    Kb_func.back(q, b, Ga.toys, toys_num_of_filters, order_toys, dict_list, 4)

```

					ІАЛЦ. 467800.007 Д4	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		